Roxen WebServer 2.2

1 Administrator Manual

2 Web Site Creator Manual

3 Tutorials

4 Programmer

5 Pike Tutorial

The inside of Internet

Table of Contents

Introduction 9
- <example-tag></example-tag> 9
- <internal/> 9

Basic Concepts 11
- URL 11
- HTTP 11
- XML 12

Roxen Concepts 13
- RXML Evaluation 13
- URL Extensions 14

Variables, Scopes & Entities 17
- Encoding 18
- Client 18
- Cookie 19
- Form 19
- Page 19
- Roxen 21
- Var 21

Information Tags 23
- <accessed/> 23
- <configurl/> 24
- <countdown/> 24
- <date/> 25
- <help/> 28
- <modified/> 28
- <number/> 29
- <roxen/> 29
- <user/> 29

Text Tags 31
- <ai></ai> 31
- <autoformat></autoformat> 31
- <case></case> 31
- <charset/> or <charset></charset> 31
- <comment></comment> 32
- <?comment ?> 32
- <default></default> 32
- <doc></doc> 32
- <foldlist></foldlist> 33
- <ft></ft> 33
- <fd></fd> 33
- <obox></obox> 33
- <random></random> 34
- <replace></replace> 34
- <smallcaps></smallcaps> 34
- <sort></sort> 34
- <strlen></strlen> 35
- <tablify></tablify> 35
- <fields></fields> 38
- <trimlines></trimlines> 38
- <wash-html></wash-html> 39

Variable Tags 41
- <append/> or <append></append> 41

<copy-scope/> 41
<dec/> 41
<define></define> 41
<attrib></attrib> 42
<contents/> 42
<inc/> 42
<insert/> 42
<insert file/> 42
<insert href/> 42
<insert realfile/> 42
<insert scopes/> 42
<insert variable/> 43
<insert variables/> 43
<scope></scope> 43
<set/> or <set></set> 43
<sprintf></sprintf> 43
<sscanf></sscanf> 44
<undefine/> 44
<unset/> 44
<use></use> 44
<vform></vform> 44
<clear/> 45
<if vform-failed></if> 45
<if vform-verified></if> 45
<reload/> 45
<verify-fail/> 45
<verify-ok/> 45
<vinput></vinput> 45
<failed></failed> 46
<verified></verified> 46

Protocol Tags 47

<aconf></aconf> 47
<apre></apre> 47
<auth-required/> 47
<email></email> 47
<attachment/> or <attachment></attachment> 48
<header/> or <header></header> 48
<signature/> 48
<expire-time/> 48
<header/> 49
<killframe/> 49
<redirect/> 49
<remove-cookie/> 49
<return/> 49
<set-cookie/> 50
<throttle/> 50

If Tags 51

<else></else> 51
<elseif></elseif> 51
<false/> 51
<if></if> 51
<if accept></if> 52
<if client></if> 52
<if clientvar></if> 52
<if config></if> 52
<if cookie></if> 52
<if date></if> 52
<if defined></if> 52
<if domain></if> 52
<if exists></if> 53
<if expr></if> 53
<if false></if> 53
<if group></if> 53
<if internal-exists></if> 53

<if ip></if> . . . 53
<if language></if> . . . 53
<if match></if> . . . 54
<if Match></if> . . . 54
<if module></if> . . . 54
<if pragma></if> . . . 54
<if prestate></if> . . . 54
<if referrer></if> . . . 54
<if sizeof></if> . . . 54
<if supports></if> . . . 55
<if time></if> . . . 55
<if true></if> . . . 55
<if user></if> . . . 55
<if variable></if> . . . 55
<if Variable></if> . . . 55
<then></then> . . . 55
<true/> . . . 55

Flow Tags . . . 57

<catch></catch> . . . 57
<cond></cond> . . . 57
<case></case> . . . 57
<default></default> . . . 57
<for></for> . . . 57
<throw></throw> . . . 57

Graphics Tags . . . 59

Color attributes . . . 59
Internal Images . . . 59
<anfang></anfang> . . . 61
<atlas></atlas> . . . 63
<country/> . . . 63
<marker/> . . . 63
<cimg/> . . . 64
<cimg-url/> . . . 66
<colorscope></colorscope> . . . 69
<configimage/> . . . 70
<diagram></diagram> . . . 70
<colors></colors> . . . 70
<data></data> . . . 71
<legend></legend> . . . 71
<xaxis/> . . . 71
<xnames></xnames> . . . 71
<yaxis/> . . . 71
<ynames></ynames> . . . 71
<gbutton></gbutton> . . . 73
<gbutton-url></gbutton-url> . . . 75
<gh></gh> . . . 78
<gtext></gtext> . . . 79
<gtext-id/> . . . 82
<gtext-url></gtext-url> . . . 85
<imgs/> . . . 87
<tablist></tablist> . . . 87
<tab></tab> . . . 88

Emit Tags . . . 91

<emit></emit> . . . 91
<emit source="atlas"></emit> . . . 91
<emit source="cimg"></emit> . . . 92
<emit source="dir"></emit> . . . 92
<emit source="fonts"></emit> . . . 93
<emit source="js-dynamic-popup"></emit> . . . 94
<emit source="js-hide-popup"></emit> . . . 94
<emit source="languages"></emit> . . . 94
<emit source="ldap"></emit> . . . 95
<emit source="path"></emit> . . . 95

<emit source="sources"></emit> 95
<emit source="sql"></emit> 95
<emit source="values"></emit> 95

Database Tags 97

<ldap/> 97
<sqlquery/> 97
<sqltable/> 97

Programming Tags 99

<cache></cache> 99
<?cdata ?> 100
<cgi/> 100
<clear-session/> 100
<crypt></crypt> 100
<debug/> 100
<dice></dice> 101
<eval></eval> 101
<force-session-id/> 101
<fsize/> 101
<gauge></gauge> 101
<maketag></maketag> 101
<attrib></attrib> 102
<nocache></nocache> 102
<nooutput></nooutput> 102
<noparse></noparse> 102
<?noparse ?> 102
<page-size/> 102
<?perl ?> 102
<?pike ?> 102
<session></session> 103
<set-max-cache/> 103
<trace></trace> 104
<writefile></writefile> 104

Script 105

CGI 105
Pike 105
Perl 105
Java 105
Javascript 106

Javascript Support 107

<js-dynamic-popup-div/> 107
<js-external></js-external> 107
<js-include/> 107
<js-insert/> 107
<js-popup></js-popup> 107
<js-write></js-write> 108

SSI Tags 109

<!--#config --> 109
<!--#echo --> 109
<!--#exec --> 110
<!--#flastmod --> 110
<!--#fsize --> 110
<!--#include --> 110
<!--#printenv --> 110
<!--#set --> 110

Compatibility 111

<counter/> 111
<cset></cset> 111
<if eval></if> 111

Security 113

.htaccess 113

.htgroup 114
.htpasswd 114

Introduction

This part of the documentation is intended for anyone who creates and publishes web pages using a Roxen WebServer. It describes the functionality Roxen WebServer provides that can be used to automate and ease the creation of both advanced static content as well as dynamic Internet applications.

Most of Roxen WebServer's functions are available as RXML tags, easily learned by anyone who knows HTML, XHTML or XML. A basic introduction is provided but the reader is encouraged to obtain more knowledge from other sources, should this information be new.

The manual is structured into four different parts. First is the introductionary part, which you have just entered, that outlines different technologies and concepts important for creating websites with Roxen products. After the general introductions follows a closer look at Roxen specific technologies with the main focus on RXML, including the uses of RXML and the syntax and semantics of the RXML language. The third part of the manual is a reference part which lists and describes all user tags. The list is grouped by the different tags general application so that it should be easy to find a tag in a solution-centric approach. Several reference chapters starts out with a closer look at some common properties of the tags in the group. The last chapter is an alphabetical list of all user tags.

The Manual Source

The reference documentation found in this manual is directly derived from the documentation in the source code of Roxen WebServer and its modules. The purpose of this approach is to get the documentation as close to the actual implementation so that is should be as easy as possible to keep it up to date. The big benefit of having the documentation in the source code is possibility to online documentaion within the product. All tag documentation found in this manual is available through the <*help*> tag as well as the documentation tab in the adminstration interface.

If you want to know more about a tag when you are developing a web page, simply add the <*help*> tag to a page and view it to see the latest documentation for that tag. Note that the module in which the tag is defined must be loaded into the server in order for this feature to work. The online documentation feature is however not restricted to only modules originating from Roxen Internet Software. Any third party or locally developed modules containing a compatible documentation may be viewed by the online documentation feature.

<example-tag></example-tag>

Provided by module: *TagDoc Notation exemplifier*

This is how the tag documentation looks like. This tag has been flagged as a container tag, i.e. you can put content into it like this: <example-tag>content</example-tag>. A tag may also be tagged as a tag-only tag, i.e. you may only write it as <example-tag/>.

Attributes

age="number"

This is the documentation of the 'age' attribute to the `<example-tag>`. In this case the attribute accepts a number, e.g. <example-tag age='42'></example-tag>. This attribute is required. If it doesn't exists in the tag you will get an RXML parse error.

sort="{up, down}" (up)

This is the documentation of the 'sort' attribute. The sort attribute may have either the value 'up' or the value 'down'. If the attribute is omitted, the tag will assume the value 'up'.

&_.ent;

Provided by *TagDoc Notation exemplifier*

This entity is an internal entity of the `<example-tag>` and only available inside it, just like `<internal>`.

<internal/>

Provided by module: *TagDoc Notation exemplifier*

This is an internal tag to `<example-tag>`, which means that it is only available inside the `<example-tag>`. Below is an example of how tag usage examples looks like. These may be a single box just showing how to write the tag or it could be a double box showing both the code and the result.

```
<example-tag><internal/></example-tag>
```

Basic Concepts

In this chapter you'll find an overview of the concepts and standards that web servers in general and Roxen WebServer and RXML in particular are built upon. These are only provided as a short introduction and background information to what follows in this manual.

We will take a look at three main topics, URLs, HTTP and XML. URLs are used to point out and locate a resource on the Internet. HTTP is used to retrieve that resource and interact with the webserver, and XML is used as the basis for describing resources. The URL section will introduce http/https/ftp URLs and outline the differences between absolute URLs, relative URLs and absolute path URLs. The HTTP section will discuss the stateless properties of the HTTP protocol and explain how this affects HTTP authentication and the need for cookies. It then ends by quickly mention the role of MIME types. Finally the XML section introduces the XML syntax and outlines the XML application XHTML. There is no mention of HTML, since those who do not know HTML are better off learning XHTML, and those who do know HTML have no problems applying XML restrictions on HTML to get XHTML.

URL

Absolute URLs

One of the most important concepts of the WWW is the URL standard, Uniform Resource Location. A URL is used as a pointer to a resource, usually a web page, on the Internet. A typical URL may look like this:

```
http://www.roxen.com/index.xml
```

This URL can be splitted up into three different parts. Everything before "://" is the protocol, in this example "http", which is the hyper text transfer protocol normally used on the WWW. Between the "://" and the first "/" is the host and the rest is the path. What is then the actual information contained in the line above? It simply means retrieve the document "/index.xml" from the host "www.roxen.com" with the HTTP protocol. Other common protocols are "https", which is secure HTTP utilizing the security layer SSL or TLS, and "ftp", which is an older way of transporting files.

The host part of the URL may optionally include a port description. If you consider the host domain as an address to the host, you can consider the port number the aparment number. The port number is implied by the selected protocol, e.g. 80 for http and 443 for https, but sometimes you want to reach a site that is not on the default port. This is done by appending ":" and the port number to the host part of the URL. The following URL points to the same resource as the above one.

```
http://www.roxen.com:80/index.xml
```

These type of URLs, containing protocol, host and path, are referred to as absolute URLs.

Relative URLs

Another often used type of URLs are the relative URLs. They refer to another document from within a document, e.g:

```
search.xml
```

The URL above means "the file called search.xml in the same directory as the document refering to it". If a link to "search.xml" was found in "http://www.roxen.com/index.xml" it means "http://www.roxen.com/search.xml". But if the same URL was found as a link in "http://www.roxen.com/platform/index.xml" it would mean "http://www.roxen.com/platform/search.xml".

A relative URL may also point to directories above or below its position. If we want to link to search.xml in a subdirectory games it would be "games/search.xml". If we on the other hand want to link to search.xml in the directory above we would use the ".." to denotate up, e.g. "../search.xml". It is possible to combine several ".." tokens, or combine them with directory names to walk down another path branch, e.g. "../../platform/search.xml".

Absolute path URL

Somewhere between absolute and relative URLs we'll find the absolute path URLs. They are not relative with respect to where on the server the URL was found, but it is relative with respect to on which server it was found.

```
/index.xml
```

The above URL means "the file index.xml in the top directory of the current server". Choosing the right URL for the right occasion will often reduce maintenance problems. A group of files that uses relative links between them can easily be moved to another directory or server. A file that links to the main search page with absolute path URLs can easily be moved to another directory or server. Luckily the shortest possible URL is often the best choice.

URLs are standardized in RFC 2396. Allowed and forbidden characters as well as characters with special meaning is of particular interest when developing Internet applications and web sites.

HTTP

The most popular protocol for transfering documents over the WWW is the hypertext transfer protocol, HTTP. Typically the browser connects to the web server sending over a request for a URL, gets a response from the server and then the connection is closed. This means that the browser has to connect to the server for every thing it downloads, e.g. if a HTML page has 40 images the browser needs to make

1+40 separate requests to the server. Since there is no persistent connection between the browser and the server there is no way to know if the user is looking at the web page just sent to him, or if she continued to look at other web pages from another server.

The original intent with HTTP was to make it a "stateless" protocol, i.e. that all requests to the server is independent, from the servers point of view. In practice that means that each server response should only rely on the information given in that very request. The benefit with this approach is that it becomes easier to make an efficient server implementation that can server web pages to any number of users, since no information about the users or the requests needs to be stored once a response has been transmitted.

Authentication

A stateless protocol has some interesting implications for authentication. As you perhaps know it is possible for a web server to respond with a "authentication required" response, telling the browser that it has to provide a user name and a password in order to get the contents at the given URL. This is often referred to as a login request. If the webserver accepts the user name and the password the browser remembers the user name/password pair as a valid authentication for that server, and will use them for all subsequent requests to the server. (This is an oversimplification. Read about authentication realms in RFC 2617) As a consequence there is, contrary to common belief, no logout mechanism. This is often emulated by temporary rejecting the valid user name/password causing the browser to drop the user name/password pair as valid authentication. The drawback is that the user will be presented with a new login request that she has to cancel.

Cookies

Another way of overcoming the drawbacks of the stateless protocol model is cookies. A cookie is a browser variable that can be set and altered by the server. Once set the browser will include the cookie in all requests to the server, thus large cookies will "waste" a lot of bandwidth. When the server sets a new value to a cookie it also gets to decide the realm of the cookie, e.g. to which URLs the browser should send the cookie, and an expiration date, at which time the browser automatically removes the cookie. There is no defined method to remove a cookie, so that operation is often simulated by setting the cookie value to an empty string and setting the expiration time to a date that has already occured. Some browsers remove the cookie at once while others wait until the next time it is restarted. Read more about cookies in RFC 2109.

Content-Type

The URL of a resource doesn't necessarily give away the type of its contents, and URLs were never intended to be used for that either. Hence every response from the web server contains a content type header containing the contents "MIME type", e.g. text/html for HTML files or image/gif for GIF files. In Roxen WebServer these MIME types are usually derived from the file extension by the content type module, but scripts and other modules may choose another MIME type. MIME types are handled by IANA and a complete list of all official MIME types can be found at www.iana.org.

XML

Tags

XML is a way of describing how to describe things. You can think of XML as the alphabet which you can use to create words with. Then you take these words and actually perform the actual act of describing things. These uses of XML are often referred to as XML applications. One of these XML applications is XHTML, which is a markup language made for use on the WWW. E.g. if I want the word "blind" to be bold in the sentence "Three blind mice." I would write

```
Three <b>blind</b> mice.
```

The and are called tags, a start tag and an end tag. The starttag, , turns bold on and the endtag, , turns off the bold property. The XML rules regarding tags are fairly simple, you must turn off, or close, all tags that you have opened and you must do it in the reverse order of how you opened them. An example with both bold and italic in XHTML:

```
<b>Three <i>blind</i></b> <i>mice</i>.
```

An empty tag <x></x> may be compressed into <x/>. This is useful for tags that doesn't need any content to be meaningful, e.g. the line break tag <br/> in XHTML.

Attributes

Tags can be made more specific by the use of attributes, which consists of an attribute name and an attribute value, or argument.

```
<b lang="en-uk">Three blind mice.</b>
```

Note that either two " or two ' may be used around the argument, depending on the contents. Example:

```
Did you hear that someone actually donated a pie to
<person name='William "Bill" Gates'>him</person>?
```

Entities

In addition to tags XML also specifies entities, which are used as a constant that represents something else. It is for example forbidden to use the characters < and > for anything else than making tags in XML. Thus there must be some other way to write these characters when you need them, < (less than) and > (greater than).

More information about both XML and XHTML is available at www.w3.org.

Roxen Concepts

Before looking into the actual features provided for static and dynamic page generation, let's have a look at what we want to accomplish with these features. The following list outlines four major properties in the WWW that must be implemented on the server side to be reliable and secure.

- Connection to content generation, such as image rendering or database retrieval.
- Interaction, such as posting information on chat boards, committing polls or performing searches in search engines.
- Monitoring and control, such as bandwidth throttling, access control and page access counters.
- Traffic control, such as HTTP cache settings and HTTP redirects.

Furthermore we want to be able to do content/layout separation as well as using macro functions to shorten development time and ease maintenance. Though these two properties does not have to be done at the server side, it is a good compromise between letting a content generation program creating static content and relying on the client to interpret the information correctly.

Scripts

Traditionally scripts has been the answer to demand for the above properties. A script is a small program that is run when a client has requested a certain URL, and the output from the program is then sent back to the client. The benefits are complete control over the transaction enabling implementation of all the above properties, but there are several drawbacks. Scripts are more expensive to handle, since you need programmers to create programs, and it is difficult to reuse the script code in an efficient manner.

Several types of scripts are supported by Roxen WebServer. Pike scripts are the most efficient, since they run inside the server process itself. The second most efficient way of using scripts is using the Extscript feature in Roxen, currently supporting Perl, which keeps the overhead of starting a Perl interpretor low by letting several requests use the same interpretor before it is discarded. Finally Roxen WebServer supports the FCGI and CGI interface making it possible to run virtually any script.

Modules

Modules in Roxen WebServer can be considered as scripts through which all requests passes, and it is up to the module to decide when to react on a request. This solves both the problems with scripts, that they are not general and that they are not as manageable. The script code can now be reused for any page that we like, e.g. by inserting a marker into the page or by explicitly adding the page URL to that modules configuration. The use of markers to trigger some kind of action in the module, e.g. inclusion of certain data, also makes the web site more manageable, since backend programmers and page designers now work on different levels.

This description of modules is an oversimplification, since there are many different kinds of modules in Roxen WebServer, with different tasks and ways to be activated. Read the Adminstrator and Programmer manual for more in-depth information.

RXML

RXML is Roxens own "marker system" for many of its modules. RXML, RoXen Makro Language (it was invented before XML), is a functional, serverside, XML compliant scripting language that easily integrates itself with web content. Since its syntax is already familiar for many it is often very easy to learn. Its tight integration with the XML in the pages enables the module programmer to keep as little HTML as possible in the program code, thus leaving more creative freedom to the web designer, i.e. a good separation between layout and functional code.

```
<table>
<emit source="sql" query="select * from users">
  <tr><td>&_.name;</
td><td><a href="mailto:&_.email;">&_.email;</a></
td></tr>
</emit>
</table>
```

In-page scripting

To "complete the circle" it should also be mentioned that it is possible to do in-page scripting, i.e. putting functional code inside the actual web pages. This is currently possible with Pike code and Perl code.

RXML Evaluation

RXML is an XML compliant programming langauge which can be used to produce dynamic as well as static content. The core of RXML is tags and entities, but unlike XML both tags and entities may be expanded into a dynamic value.

To make it easier to use entities several entities are grouped into something called a scope. You can think of a scope as a bucket with variables, so when you reference a variable with an entity you first select the right bucket and then the right variable. In the example below the variable thing in the scope var is set to "Book" and the variable thing in the scope form is set to "Chair". In the last row of the example these to variables are inserted into the document.

```
<set variable='var.thing'>Book</set>
<set variable='form.thing'>Chair</set>
My &var.thing; is on the &form.thing;.

My Book is on the Chair.
```

As a rule of thumb entities are expanded first and the tags are evaluated later, as is exemplified in the next example.

First var.thing is set to "Book" and then form.thing is set to the value in var.thing, i.e. "Book".

```
<set variable='var.thing'>Book</set>
<set variable='form.thing'>&var.thing;</set>
My &var.thing; is on the &form.thing;.

My Book is on the Book.
```

You can also use entities to insert values into the attributes of tags, exemplified by these two examples.

```
<set variable='var.variable'>var.title</set>
<set variable='&var.variable;'>Autour de la Lune</
set>
Title: &var.title;

Title: Autour de la Lune
```

```
<set variable='var.variable'>title</set>
<set variable='var.&var.variable;'>Autour de la Lun
e</set>
Title: &var.title;

Title: Autour de la Lune
```

Evaluation order

The general evaluation order of RXML is from top to bottom of the page, and inside and out in terms of tag levels. The "top to bottom" means that if two unnested tags appears on a page, the first one on the page is evaluated before the second one.

```
<set variable='var.value'>One</set>
&var.value;<br />
<append variable='var.value'> Two</append>
&var.value;<br />

One<br />
One Two<br />
```

The "inside and out" on the other hand describes how tags behave if they are nested. Then the innermost first gets to produce its result. That result is then handed over to the embracing tag, which in turn produces its output with the first tags output as input.

```
<case case='upper'><date part='wday' type='string'
/></case>

MONDAY
```

URL Extensions

Since everything after the host part of the URL is sent to the server as is, after transport encoding, the server in practice decides the meaning of that part of the URL. Normally that part of the URL consists of a path part, containing the path to a file the client wants, and a query part, containing variables that may affect the way the server accomplishes the request.

```
http://roxen.com/products/platform/tech-
features.xml?page=9
```

In the URL above we request the file /products/platform/tech-features.xml from the host roxen.com with the variable page set to 9. Note that there is in principle nothing that prevents us from making a server that returns the same result with the following URL.

```
http://roxen.com/
SELECT_page_FROM_features_WHERE_product=platform_AN
D_page=9
```

This is however not a good idea for several reasons, usability being the number one. Users are used to alter the URL of a page to get to index-pages higher up in the web site structure.

Index pages

The most common sidestep from the rule that the path part of the URL explicitly denotes a file is directory URLs.

```
http://roxen.com/products/platform/
```

The above URL does not denotes a specific file in the /products/platform/ directory, but does instead point at the directory itself. The common approach is to find an index file in the directory and send that file instead. This is handled by the directory module, which by default looks for the files "index.html", "index.xml", "index.htm", "index.pike" and "index.cgi" in that order.

Path info

Somtimes it can be practical to fake a directory structure, but let all requests to the files in that directory lead to the same file. The example with the tech-features.xml URL above could look like this:

```
http://roxen.com/products/platform/tech-
features.xml/9/
```

The part of the URL after the actual file will then be provided to the file/script in a special variable during its parsing.

Prestates

When developing and debugging is a great help to be able to turn on and off specific parts of the code that generates the current page. This is an ideal application for prestates, a mechanism invented by Roxen to enable the user to turn certain switches on and off. The name and function of the prestates is decided by the page developer. One example of how prestates are used is the Table/Image Border Unveiler module, which is used on the community.roxen.com web site.

```
http://community.roxen.com/(tables)/developers/
```

This URL signifies that we want to fetch file at the path /developers/ from the host commuinity.roxen.com with the protocol HTTP and with the prestate tables set. In the WebServer the Table/Image Border Unveiler module recognizes the table prestate and knows that the user wants all tables highlighted in the page. Compare the result with how the page look without the prestate. It is also possible to add several prestates to the same page in a comma separated list, e.g.

```
http://community.roxen.com/(tables,images)/
developers/
```

Prestates can of course be used for many other things than switching debugging flags, e.g. moving states between pages like a browser window local cookie. See *<apre>* and *<if prestate>* for more information about how to control and detect prestates in your RXML applications.

Config states

A variation of prestates is the config state. Looks very similar to the prestates, but stores its value in a cookie. Looking at the following URL will store the value "bacon" in the cookie RoxenConfig, which will be valid for two years since its latest change. After the cookie has been set, the server will redirect to the page you came from, or it it was unable to dermine that page, to the same URL but without the config state.

```
http://community.roxen.com/<bacon>/
```

Removing a config state value is a little trickier than with prestates, since you can not edit them by hand, as with the URL. Prepending a minus sign before a config state flag indicates that it should be removed from the RoxenConfig cookie. As with prestates it is possible to combine several states at the same time, both with and without minus signs.

```
http://community.roxen.com/<egg,-bacon>/
```

See *<aconf>* and *<if config>* for more information about how to control and detect config stats in your RXML applications.

Variables, Scopes & Entities

An entity is the collective name for a variable that in RXML 2 has been given an all accessible name, which mean it is available everywhere whenever the module defining it is loaded. A more correct name might be *variable-entity*, but in RXML 2 it is referred to as an *entity*.

An entity mostly contains information fetched from within the Roxen WebServer, but it can also be declared by a tag, making it a carrier of website specific information.

A variable always belongs to a scope, i.e. a group of variables providing information about something specific, e.g. a *client* or a *page*. A list of variables belonging to a scope can be made available by this RXML code:

```
<pre>
<insert variables="full" scope="the scope"/>
</pre>
```

Entity syntax

The syntax that specifies a variable as an entity is "scope.variable", e.g. page.filename.

Quoting entities

When outputting information from a source that contains entities that should be parsed after the output is parsed, it is necessary to quote these entities else they will be parsed in advance. Quoting entities is done by adding an extra ":" after the opening ampersand, e.g. :scope.variable.

Scopes

The most common scopes that handles variables are the *Var* and *Form* scopes. The *Var Scope* is always empty when the page parsing begins and should be used for temporary variables. The *Form Scope* contains all returned results from forms.

There is no default scope on the page top level, i.e. outside of all tags that creates scopes, unless you are running in compatibility mode. Then the *Form Scope* will be your default scope. The scope "_" (underscore) is always the present scope, e.g. in *<emit sql>* where sql.x = _.x.

```
<emit source="sql" query=" ... " scope="outer">
 &_.name; = &outer.name;
  <emit source="sql" query=" ... " scope="inner">
   &_.name; = &inner.name; != &outer.name;
  </emit>
</emit>
```

This is an example on how entities works inside nestled *<emit>* tags. Notice how the present scope "_" (underscore) changes from the first `<emit>` to the second.

Entities in RXML 2 are handled like any variable in RXML 1.3. Examples:

RXML	Result
`<set variable="var.foo" value="bar"/>`	var.foo = bar
`<set variable="var" scope="foo" value="bar"/>`	foo.var = bar
`<if variable="var.foo = bar">gazonk</if>`	Returns gazonk if var.foo = bar.
`<if match="var.foo = *test">gazonk</if>`	Returns gazonk if var.foo ends in test.

Splice

The splice attribute-operand '::', has a similar function in RXML as in most programming languages. Splice in RXML is used for expanding what is inside a variable, i.e. put what is stored in the variable into for instance a tag before parsing it. It might for instance be a list of attributes stored in the variable or an URL, etc. The splice operand is necessary as the RXML-parser (RXML 2) only will make one pass when parsing a page while the old parser (RXML 1.3) sometimes made several passes while parsing a tag.

```
<define variable='var.foo'>quant='20' gamma='0.5'</
define>
<cimg src='../img/testimage.jpg' ::='&var.foo;'/>
<cimg src='../img/
testimage.jpg' quant='20' gamma='0.5'/>
```

The really useful thing with :: is that it is possible to give attributes, which are unknown, to a tag in advance, which makes it is possible to store an entire list of attributes in a variable.

```
<sandwich ::="bread='rye' &var.stuffings;" butter="
low-fat"/>
&lt;sandwich bread='rye' lettuce="" ham="" tomato="
" butter="low-fat"/&gt;
```

Encoding

An entity can be placed anywhere in a page and will have its content inserted during the parsing. The content is encoded by default so that "<" will be output as "<" in HTML pages. You can choose another scheme by using the *scope.variable:scheme* syntax, e.g. form.html:none. *[More about encoding]*

The scopes are:

Encoding

All variables in RXML 2 are accessed through entities, e.g. var.foo.

By default, an entity will be HTML encoded, that is, **<** will be inserted as **<**, **>** as **>** and **&** as **&**. However, there are instances when that is not what you want, for example, when inserting entities into SQL queries. Therefore, the encoding can be controlled by applying another encoding scheme on the entity, scope.entity:*scheme*.

```
<sqlquery query="SELECT * FROM db WHERE name='&form
.name:mysql;'">
```

Available Encoding Schemes

- No quoting. This is dangerous and should never be used unless you have total control over the contents of the variable.
- The default quoting, for inserting into regular HTML or RXML. Encoded characters are **&**, **<**, **>**, **"**, **'** and the null character.
- For inserting variables into URLs. Encoded characters are the null character, space, tab, the newline character, the carriage return character, **%**, **'**, **"**, **#**, **&**, **?**, **=**, **/** and **:**.
- Uses a subset of the URL encoding scheme. Characters **&** and **?** are not encoded as it would make inserting i.e. variables into http-strings impossible. Encoded characters are the null character, space, tab, the newline character, the carriage return character, **%**, **'** and **"**.
- Uses a subset of the URL encoding scheme. Only the characters **=**, **,**, **;** and **%** are encoded.
- For inserting into Pike strings, for use with the `<pike>` tag. Encoded characters are **"**, **** and the newline character.
- For inserting into Javascript strings. Encoded characters are the backspace character, the formfeed character, the newline character, the return carriage character, tab, ****, **'** and **"**.
- For inserting into MySQL SQL queries. Encoded characters are **"**, **'** and ****.
- For inserting into MySQL SQL queries in pike strings. Encoded characters are **"**, **'**, **** and the newline character.
- For inserting into SQL queries. Encoded character is **'**.

Client

This scope contains information specific to the client/browser that is accessing the page. All support variables defined in the support file is added to this scope.

&client.Fullname;
Provided by *Tags: RXML 2 tags*
The full user agent string, i.e. name of the client and additional info like; operating system, type of computer, etc. An example output: "Mozilla/4.7 [en] (X11; I; SunOS 5.7 i86pc)".

&client.accept-language;
Provided by *Tags: RXML 2 tags*
The client prefers to have the page contents presented in this language, according to the accept-language header. An example output: "en".

&client.accept-languages;
Provided by *Tags: RXML 2 tags*
The client prefers to have the page contents presented in these languages, according to the accept-language header. An example output: "en, sv".

&client.authenticated;
Provided by *Tags: RXML 2 tags*
Returns the name of the user logged on to the site, i.e. the login name, if any exists.

&client.fullname;
Provided by *Tags: RXML 2 tags*
The full user agent string, i.e. name of the client and additional info like; operating system, type of computer, etc. Unlike client.fullname this value is lowercased. An example output: "mozilla/4.7 [en] (x11; i; sunos 5.7 i86pc)".

&client.height;
Provided by *Tags: RXML 2 tags*
The presentation area height in pixels. For WAP-phones.

&client.host;
Provided by *Tags: RXML 2 tags*
The host name of the client, if possible to resolve. An example output: "www.roxen.com".

&client.ip;
Provided by *Tags: RXML 2 tags*
The client is located on this IP-address. An example output: "194.52.182.15".

&client.javascript;
Provided by *Tags: RXML 2 tags*
Returns the highest version of javascript supported.

&client.language;
Provided by *Tags: RXML 2 tags*
The clients most preferred language. Usually the same value as client.accept-language, but is possibly altered by a customization module like the Preferred language analyzer. It is recommended that this entity is used over the client.accept-language when selecting languages. An example output: "en".

&client.languages;
Provided by *Tags: RXML 2 tags*
An ordered list of the clients most preferred languages. Usually the same value as client.accept-language, but is possibly altered by a customization module like the Preferred language analyzer, or reorganized according to quality identifiers according to the HTTP specification. An example output: "en, sv".

&client.name;
Provided by *Tags: RXML 2 tags*
The name of the client, i.e. the sent user agent string up until the first space character. An example output: "Mozilla/4.7".

&client.password;
Provided by *Tags: RXML 2 tags*
Returns the password the user used when he/she tried to log on the site.

&client.referrer;
Provided by *Tags: RXML 2 tags*
Prints the URL of the page on which the user followed a link that brought her to this page. The information comes from the referrer header sent by the browser. An example output: "http://www.roxen.com/index.xml".

&client.robot;
Provided by *Tags: RXML 2 tags*
Returns the name of the webrobot. Useful if the robot requesting pages is to be served other contents than most visitors. Use client.robot together with *<if>*.
Possible webrobots are: ms-url-control, architex, backrub, checkbot, fast, freecrawl, passagen, gcreep, getright, googlebot, harvest, alexa, infoseek, intraseek, lycos, webinfo, roxen, altavista, scout, slurp, urlminder, webcrawler, wget, xenu and yahoo.

&client.session;
Provided by *Session tag module*
Contains a session key for the user or nothing. The session key is primary taken from the RoxenUserID cookie. If there is no such cookie it will return the value in the prestate that begins with "RoxenUserID=". However, if both the cookie and such a prestate exists the client.session variable will be empty. This allows the client.session variable to be used together with `<force-session-id>`. Note that the Session tag module must be loaded for this entity to exist.

&client.tm;
Provided by *Tags: RXML 2 tags*
Generates a trademark sign in a way that the client can render. Possible outcomes are "™", "^{TM}", and ">TM<".

&client.user;
Provided by *Tags: RXML 2 tags*
Returns the name the user used when he/she tried to log on the site, i.e. the login name, if any exists.

&client.width;
Provided by *Tags: RXML 2 tags*
The presentation area width in pixels. For WAP-phones.

Cookie

This scope contains the cookies sent by the client. Adding, deleting or changing in this scope updates the clients cookies. There are no predefined entities for this scope. When adding cookies to this scope they are automatically set to expire after two years.

Form

This scope contains the form variables, i.e. the answers to HTML forms sent by the client. Both variables resulting from POST operations and GET operations gets into this scope. There are no predefined entities for this scope.

Page

This scope contains information specific to this page.

&page.accessed;
Provided by *Tags: Accessed counter*
Generates an access counter that shows how many times the page has been accessed. Needs the accessed module.

&page.author;
Provided by *SiteBuilder: Tags 2.0*
Returns the author's name.

&page.author-id;
Provided by *SiteBuilder: Tags 2.0*
Returns the author's identification number.

&page.author-name;
Provided by *SiteBuilder: Tags 2.0*
Returns the author's handle, i.e. login name.

&page.content-editor;
Provided by *SiteBuilder: Tags 2.0*
Returns the full internal SiteBuilder URL to the page in the Content Editor, focusing on the file.

&page.description;
Provided by *SiteBuilder: Tags 2.0*
Returns the description of the file, set in the metadata.

&page.dir;
Provided by *Tags: RXML 2 tags*
The name of the directory in the virtual filesystem where the file resides, as derived from the URL. If the URL is "http://community.roxen.com/articles/index.html", then the value of this entity is "/articles/".

&page.filename;
Provided by *SiteBuilder: Tags 2.0*
Returns the filename.

&page.filesize;
Provided by *Tags: RXML 2 tags*
This file's size, in bytes.

&page.filesize;
Provided by *SiteBuilder: Tags 2.0*
Returns the filesize.

&page.keywords;
Provided by *SiteBuilder: Tags 2.0*
Returns the keywords of the file, set in the metadata.

&page.language;
Provided by *Tags: RXML 2 tags*
What language the contents of this file is written in. The language must be given as metadata to be found.

&page.languages;
Provided by *SiteBuilder: Tags 2.0*
All languages present in the file returned as a comma-separated string.

&page.last-true;
Provided by *Tags: RXML 2 tags*
Is "1" if the last `<if>`-statement succeeded, otherwise 0. (*<true>* and *<false>* is considered as `<if>`-statements here) See also: *If Tags*.

&page.modification-date;
Provided by *SiteBuilder: Tags 2.0*

Returns the date when the file was last modified.

&page.modification-time;
Provided by *SiteBuilder: Tags 2.0*
Returns the time when the file was last modified.

&page.path;
Provided by *Tags: RXML 2 tags*
Absolute path to this file in the virtual filesystem. E.g. with the URL "http://www.roxen.com/partners/../products/index.xml", as well as "http://www.roxen.com/products/index.xml", the value will be "/products/index.xml", given that the virtual filsystem was mounted on "/".

&page.pathinfo;
Provided by *Tags: RXML 2 tags*
The "path info" part of the URL, if any. Can only get set if the "Path info support" module is installed. For details see the documentation for that module.

&page.permission;
Provided by *SiteBuilder: Tags 2.0*
Returns the string "read" or "write" depending on whether the viewer has permission to write in the file or not.

&page.query;
Provided by *Tags: RXML 2 tags*
The query part of the page URL. If the page URL is "http://www.server.com/index.html?a=1&b=2" the value of this entity is "a=1&b=2".

&page.realfile;
Provided by *Tags: RXML 2 tags*
Path to this file in the file system. An example output: "/home/joe/html/index.html".

&page.revision;
Provided by *SiteBuilder: Tags 2.0*
Returns the file's current revision number.

&page.scope;
Provided by *Tags: RXML 2 tags*
The name of the current scope, i.e. the scope accessible through the name "_".

&page.selectable;
Provided by *SiteBuilder: Tags 2.0*
For showing whether an XSL template file is selectable within the "Edit metadata" wizard. Returns either "n/a" if the file is of an other content-type than "site-builder/xsl-template", "yes" if the file is selectable or "no" if the file isn't selectable.

&page.selected;
Provided by *SiteBuilder: Tags 2.0*
Whether this file is the current file or if this directory is a directory within the path to the current file. Only one entry will be selected.

&page.self;
Provided by *Tags: RXML 2 tags*
The name of this file, derived from the URL. If the URL is "http://community.roxen.com/articles/index.html", then the value of this entity is "index.html".

&page.site-status;
Provided by *SiteBuilder: Tags 2.0*
Returns the file's status in the site's workarea.

&page.ssl-strength;
Provided by *Tags: RXML 2 tags*
The number of bits used in the key of the current SSL connection.

&page.stationery;
Provided by *SiteBuilder: Tags 2.0*
Returns "yes" if the page is marked as a stationery, "no" otherwise.

&page.status-img;
Provided by *SiteBuilder: Tags 2.0*
Returns the full internal SiteBuilder URL to the page's current status image. The status image tells if the page has been modified etc. from the viewer's point of view. This is the same status icon as used by the Content Editor.

&page.template;
Provided by *SiteBuilder: Tags 2.0*
Returns the current page's selected template.

&page.title;
Provided by *SiteBuilder: Tags 2.0*
Returns the title of the file, set in the metadata.

&page.type;
Provided by *SiteBuilder: Tags 2.0*
Returns the file's content-type.

&page.type-img;
Provided by *SiteBuilder: Tags 2.0*
Returns the internal SiteBuilder URI to where the file's content-type image is stored.

&page.url;
Provided by *Tags: RXML 2 tags*
The absolute path for this file from the web server's root view including query variables.

&page.user-status;
Provided by *SiteBuilder: Tags 2.0*
Returns the file's status in the user's workarea.

&page.virtroot;
Provided by *Tags: RXML 2 tags*
The root of the present virtual filesystem, usually "/".

&page.visible;
Provided by *SiteBuilder: Tags 2.0*
Returns "yes" if the page is currently externally visible, "no" otherwise.

&page.visible-from;
Provided by *SiteBuilder: Tags 2.0*
Returns the date and time, in iso format, when the page starts to be visible. If external visibility is set to *Never*, "never" will be returned. If external visibility is set to *Visible util a specified time*, "now" will be returned.

&page.visible-to;
Provided by *SiteBuilder: Tags 2.0*
Returns the date and time, in iso format, when the page ends to be visible. If external visibility is set to *Never*, "never" will be returned. If external visibility is set to *Visible after a specified time*, "indefinite" will be returned.

&page.workarea;
Provided by *SiteBuilder: Tags 2.0*
Returns the name of the workarea where the viewed page is stored.

&page.workarea-id;
Provided by *SiteBuilder: Tags 2.0*
Returns the unique id of the workarea. Useful when doing web applications.

Roxen

This scope contains information specific to this Roxen WebServer. It is not possible to write any information to this scope.

&roxen.domain;
Provided by *Tags: RXML 2 tags*
The domain name of this site. The information is taken from the client request, so a request to "http://community.roxen.com/" would give this entity the value "community.roxen.com", while a request for "http://community/" would give the entity value "community".

&roxen.hits;
Provided by *Tags: RXML 2 tags*
The number of hits, i.e. requests the webserver has accumulated since it was last started.

&roxen.hits-per-minute;
Provided by *Tags: RXML 2 tags*
The average number of requests per minute since the webserver last started.

&roxen.pike-version;
Provided by *Tags: RXML 2 tags*
The version of Pike the webserver is using, e.g. "Pike v7.2 release 140".

&roxen.sent;
Provided by *Tags: RXML 2 tags*
The total amount of data the webserver has sent since it last started.

&roxen.sent-kbit-per-second;
Provided by *Tags: RXML 2 tags*
The average amount of data the webserver has sent, in Kibibits.

&roxen.sent-mb;
Provided by *Tags: RXML 2 tags*
The total amount of data the webserver has sent, in Mebibits.

&roxen.sent-per-minute;
Provided by *Tags: RXML 2 tags*
The average number of bytes that the webserver sends during a minute. Based on the sent amount of data and uptime since last server start.

&roxen.server;
Provided by *Tags: RXML 2 tags*
The URL of the webserver. The information is taken from the client request, so a request to "http://community.roxen.com/index.html" would give this entity the value "http://community.roxen.com/", while a request for "http://community/index.html" would give the entity the value "http://community/".

&roxen.ssl-strength;
Provided by *Tags: RXML 2 tags*
Contains the maximum number of bits encryption strength that the SSL is capable of. Note that this is the server side capability, not the client capability. Possible values are 0, 40, 128 or 168.

&roxen.time;
Provided by *Tags: RXML 2 tags*
The current posix time. An example output: "244742740".

&roxen.unique-id;
Provided by *Tags: RXML 2 tags*
Returns a unique id that can be used for e.g. session identification. An example output: "7fcda35e1f9c3f7092db331780db9392". Note that a new id will be generated every time this entity is used, so you need to store the value in another variable if you are going to use it more than once.

&roxen.uptime;
Provided by *Tags: RXML 2 tags*
The total uptime of the webserver since last start, in seconds.

&roxen.uptime-days;
Provided by *Tags: RXML 2 tags*
The total uptime of the webserver since last start, in days.

&roxen.uptime-hours;
Provided by *Tags: RXML 2 tags*
The total uptime of the webserver since last start, in hours.

&roxen.uptime-minutes;
Provided by *Tags: RXML 2 tags*
The total uptime of the webserver since last start, in minutes.

Var

General variable scope. This scope is always empty when the page parsing begins and is therefore suitable to use as storage for all variables used during parsing.

Information Tags

Information tags are simple tags that provide information about the client, the server or some external event. Examples are *<accessed>*, that counts accesses to the page and *<modified>*, which shows when the page was last updated.

<accessed/>

Provided by module: *Tags: Accessed counter*

Generates an access counter that shows how many times the page has been accessed. A file, AccessedDB, in the logs directory is used to store the number of accesses to each page. By default the access count is only kept for files that actually contain an accessed-tag, but can also be configured to count all files of a certain type.

```
<accessed/>
12
```

Attributes

add="number"
Increments the number of accesses with this number instead of one, each time the page is accessed.

addreal
Prints the real number of accesses as an HTML comment. Useful if you use the cheat attribute and still want to keep track of the real number of accesses.

case="{upper, lower, capitalize}"
Sets the result to upper case, lower case or with the first letter capitalized.

cheat="number"
Adds this number of accesses to the actual number of accesses before printing the result. If your page has been accessed 72 times and you add `<accessed cheat='100'>` the result will be 172.

database
Works like the since attribute, but counts from the day the first entry in the entire accessed database was made.

factor="percent"
Multiplies the actual number of accesses by the factor. E.g. `<accessed factor='50'>` displays half the actual value.

file="filename"
Shows the number of times the page filename has been accessed instead of how many times the current page has been accessed. If the filename does not begin with "/", it is assumed to be a URL relative to the directory containing the page with the accessed tag. Note, that you have to type in the full name of the file. If there is a file named tmp/index.html, you cannot shorten the name to tmp/, even if you've set Roxen up to use index.html as a default page. The filename refers to the virtual filesystem.

One limitation is that you cannot reference a file that does not have its own `<accessed>` tag. You can use `<accessed silent='1'>` on a page if you want it to be possible to count accesses to it, but don't want an access counter to show on the page itself.

lang="langcodes"
Will print the result as words in the chosen language if used together with type=string.

```
<accessed type="string"/>
twelve
```

```
<accessed type="string" lang="sv"/>
tolv
```

per="{second, minute, hour, day, week, month, year}"
Shows the number of accesses per unit of time.

```
<accessed per="week"/>
0
```

prec="number"
Rounds the number of accesses to this number of significant digits. If prec=2 show 12000 instead of 12148.

reset
Resets the counter. This should probably only be done under very special conditions, maybe within an `<if>` statement. This can be used together with the file argument, but it is limited to files in the current- and sub-directories.

silent
Print nothing. The access count will be updated but not printed. This option is useful because the access count is normally only kept for pages with actual `<access>` on them. `<accessed file='filename'>` can then be used to get the access count for the page with the silent counter.

since
Inserts the date that the access count started. The language will depend on the *lang* attribute, default is English. All normal date related attributes can be used. Also see: *<date>*.

```
<accessed since="1"/>
November 2001
```

type="{number, string, roman, iso, discordian, stardate, mcdonalds, linus, ordered}"
Specifies how the count are to be presented. Some of these are only useful together with the since attribute.

```
<accessed type="roman"/>
```

```
XII
```

```
<accessed since="1" type="iso"/>
2001-11-22T17:09:01
```

```
<accessed since="1" type="discordian"/>
Sweetmorn, the 33rd day of The Aftermath
```

```
<accessed since="1" type="stardate"/>
37215.8
```

```
<accessed type="mcdonalds"/>
More than ten served.
```

```
<accessed type="linus"/>
12 since Thu Nov 22 17:09:00 2001
```

```
<accessed type="ordered"/>
12th
```

minlength="number"
Defines a minimum length the the resulting string should have. If it is shorter it is padded from the left with the padding value. Only values between 2 and 10 are valid.

padding="character" (0)
The padding that the minlength function should use.

<configurl/>

Provided by module: *Tags: RXML 2 tags*

Returns a URL to the administration interface.

<countdown/>

Provided by module: *Tags: Countdown*

This tag can count days, minutes, months, etc. from a specified date or time. It can also give the time to or from a few special events. See below for a full list.

Attributes

Time:

year="number"
Sets the year to count down to.

month="{number, month_name}"
Sets the month to count down to. If given as a number January is 1.

day="{number, day_name}"
Sets the weekday to count down to. If given as a number Sunday is 1.

mday="number"
Sets the day of the month to count down to.

hour="number"
Sets the hour to count down to.

minute="number"
Sets the minute to count down to.

second="number"
Sets the second to count down to.

iso="year-month-day"
Sets the year, month and day to count down to. (YYYY-MM-DD, YYYYMMDD or YYYY-MMM-DD).

```
<countdown iso='2020-FEB-12'/>
6423 days
```

event="easter,gregorian-easter,julian-easter,christmas,christmas-day,christmas-eve"
Sets the time of an event to count down to.

years="number"
Add this number of years to the result.

months="number"
Add this number of months to the result.

weeks="number"
Add this number of weeks to the result.

days="number"
Add this number of days to the result.

hours="number"
Add this number of hours to the result.

beats="number"
Add this number of beats to the result.

minutes="number"
Add this number of minutes to the result.

seconds="number"
Add this number of seconds to the result.

now="year-month-day"
Sets the 'present' time, if other than really present time. (YYYY-MM-DD, YYYYMMDD or YYYY-MMM-DD)

```
<countdown now="1999-12-
24" year="2000" display="days"/>
8
```

Presentation:

display="{when, years, months, weeks, days, hours, beats, minutes, seconds, combined, dogyears, boolean}"

display=when
Shows when the time will occur. All arguments that are valid in <*date*> can be used to modify the display.

display=years
How many years until the time.

display=months
How many months until the time.

display=weeks
How many weeks until the time.

display=days
How many days until the time.

display=hours
How many hours until the time.

```
<countdown day='friday' display='hours'/>
0
```

display=beats
How many beats until the time.

display=minutes
How many minutes until the time.

display=seconds
How many seconds until the time.

display=combined
Shows an english text describing the time period. Example: 2 days, 1 hour and 5 seconds. You may use the 'prec' attribute to limit how precise the description is. Also, you can use the 'month' attribute if you want to see years/months/days instead of years/weeks/days.

```
  The world will go under in <countdown year='20
38' display='combined' prec='day'/>.
  The world will go under in 35 years, 5 months
and 20 days.
```

display=dogyears
How many dog-years until the time. (With one decimal)

```
<countdown years='2' display='dogyears'/>
14.0
```

display=boolean
Return true or false (1 or 0), depending on if the time is now or not. The fuzziness of 'now' is decided by the 'prec' option.

```
  <b>Is this a Sunday?</b>
<define variable='var.test' preparse=''><countdo
wn day='sunday' display='boolean'/></define>
<if variable='var.test = 1'>Yes, this is a Sunda
y.</if>
<else>No, it isn't.</else>
  <b>Is this a Sunday?</b>


No, it isn't.
```

type="type"
As for 'date'. Useful values for type include string, number and ordered.

lang="langcodes"
The language in which the result should be written if the type is string.

```
  Heute ist es ungef‰hr <countdown event='christmas
' display='months' type='string' lang='de'/
> Monate bis Weinachten.
  Heute ist es ungef‰hr f¸nf Monate bis Weinachten.
```

since
Negate the period of time.

```
  I am <countdown iso='1980-06-
28' since='' display='years' type='string'/
> years old.
  I am twentytwo years old.
```

next
Always count down to the next event. `<countdown day='friday' next=''>` says 6 on a friday as opposed to 0 without the next attribute.

```
  It is <countdown day='monday' next=''/
> to monday.
  It is 2 days to monday.
```

prec="{year, month, week, day, hour, minute, second}"
Modifies the precision for 'boolean'- and 'combined'-arguments.

<date/>

Provided by module: *Tags: RXML 2 tags*

Inserts the time and date. Does not require attributes.

```
<date/>
18:37, July the 12th, 2002
```

Attributes

unix-time="number of seconds"
Display this time instead of the current. This attribute uses the specified Unix 'time_t' time as the starting time (which is *01:00, January the 1st, 1970*), instead of the current time. This is mostly useful when the `<date>` tag is used from a Pike-script or Roxen module.

```
<date unix-time='120'/>
01:02, January the 1st, 1970
```

timezone="{local, GMT}" (local)
Display the time from another timezone.

years="number"
Add this number of years to the result.

```
<date date='' years='2'/>
July the 12th in the year of 2004
```

months="number"
Add this number of months to the result.

```
<date date='' months='2'/>
September the 12th in the year of 2002
```

weeks="number"
Add this number of weeks to the result.

```
<date date='' weeks='2'/>
July the 26th in the year of 2002
```

days="number"
Add this number of days to the result.

hours="number"
Add this number of hours to the result.

```
<date time='' hours='2' type='iso'/>
20:37:52
```

beats="number"
Add this number of beats to the result.

```
<date time='' beats='10' type='iso'/>
18:52:16
```

minutes="number"
Add this number of minutes to the result.

seconds="number"
Add this number of seconds to the result.

adjust="number"
Add this number of seconds to the result.

brief
Show in brief format.

```
<date brief=''/>
today, 18:37
```

time
Show only time.

```
<date time=''/>
18:37
```

date
Show only date.

```
<date date=''/>
July the 12th in the year of 2002
```

type="{string, ordered, iso, discordian, stardate, number, unix}"
Defines in which format the date should be displayed in. Discordian and stardate only make a difference when not using part. Note that type='stardate' has a separate companion attribute, prec, which sets the precision.

type=discordian	`<date date='' type='discordian'/>` `Pungenday, the 46th day of Confusion`
type=iso	`<date date='' type='iso'/>` `2002-07-12`
type=number	`<date date='' type='number'/>` `July the 12th in the year of 2002`
type=ordered	`<date date='' type='ordered'/>` `July the 12th in the year of 2002`
type=stardate	`<date date='' type='stardate'/>` `37447.5`
type=string	`<date date='' type='string'/>` `July the 12th in the year of 2002`
type=unix	`<date date='' type='unix'/>` `1026491872`

part="{year, month, day, wday, date, mday, hour, minute, second, yday, beat, week, seconds}"
Defines which part of the date should be displayed. Day and wday is the same. Date and mday is the same. Yday is the day number of the year. Seconds is unix time type. Only the types string, number and ordered applies when the part attribute is used.

Part	Meaning

year	Display the year.
	`<date part='year' type='number'/>`
	`2002`

month	Display the month.
	`<date part='month' type='ordered'/>`
	`7th`

day	Display the weekday, starting with Sunday.
	`<date part='day' type='ordered'/>`
	`6th`

wday	Display the weekday. Same as 'day'.
	`<date part='wday' type='string'/>`
	`Friday`

date	Display the day of this month.
	`<date part='date' type='ordered'/>`
	`12th`

Part	Meaning

mday	Display the number of days since the last full month.
	
	`<date part='mday' type='number'/>`
	`12`

hour	Display the numbers of hours since midnight.
	
	`<date part='hour' type='ordered'/>`
	`18th`

minute	Display the numbers of minutes since the last full hour.
	
	`<date part='minute' type='number'/>`
	`37`

second	Display the numbers of seconds since the last full minute.
	
	`<date part='second' type='string'/>`
	`fiftytwo`

yday	Display the number of days since the first of January.
	
	`<date part='yday' type='ordered'/>`
	`192nd`

beat	Display the number of beats since midnight Central European Time(CET). There is a total of 1000 beats per day. The beats system was designed by Swatch as a means for a universal time, without time zones and day/night changes.
	
	`<date part='beat' type='number'/>`
	`@734`

week	Display the number of the current week.
	
	`<date part='week' type='number'/>`
	`28`

Part	Meaning

seconds	Display the total number of seconds this year.
	
	`<date part='seconds' type='number'/>`
	`1026491872`

strftime="string"

If this attribute is given to date, it will format the result according to the argument string.

Format	Meaning
%%	Percent character
%a	Abbreviated weekday name, e.g. "Mon"
%A	Weekday name
%b	Abbreviated month name, e.g. "Jan"
%B	Month name
%c	Date and time, e.g. "%a %b %d %H:%M:%S %Y"
%C	Century number, zero padded to two charachters.
%d	Day of month (1-31), zero padded to two characters.
%D	Date as "%m/%d/%y"
%e	Day of month (1-31), space padded to two characters.
%H	Hour (24 hour clock, 0-23), zero padded to two characters.
%h	See %b
%I	Hour (12 hour clock, 1-12), zero padded to two charcters.
%j	Day numer of year (1-366), zero padded to three characters.
%k	Hour (24 hour clock, 0-23), space padded to two characters.
%l	Hour (12 hour clock, 1-12), space padded to two characters.

Format	Meaning
%m	Month number (1-12), zero padded to two characters.
%M	Minute (0-59), zero padded to two characters.
%n	Newline
%p	"a.m." or "p.m."
%r	Time in 12 hour clock format with %p
%R	Time as "%H:%M"
%S	Seconds (0-61), zero padded to two characters.
%t	Tab
%T	Time as "%H:%M:%S"
%u	Weekday as a decimal number (1-7), 1 is Sunday.
%U	Week number of year as a decimal number (0-53), with sunday as the first day of week 1, zero padded to two characters.
%V	ISO week number of the year as a decimal number (1-53), zero padded to two characters.
%w	Weekday as a decimal number (0-6), 0 is Sunday.
%W	Week number of year as a decimal number (0-53), with sunday as the first day of week 1, zero padded to two characters.
%x	Date as "%a %b %d %Y"
%X	See %T
%y	Year (0-99), zero padded to two characters.
%Y	Year (0-9999), zero padded to four characters.

```
<date strftime="%B %e %Y, %A %T"/>
July 12 2002, Friday 18:37:52
```

lang="langcode"
Defines in what language a string will be presented in. Used together with *type=string* and the *part* attribute to get written dates in the specified language.

```
<date part='day' type='string' lang='de'/>
Freitag
```

case="{upper, lower, capitalize}"
Changes the case of the output to upper, lower or capitalize.

```
<date date='' lang='&client.language;' case='uppe
r'/>
JULY THE 12TH IN THE YEAR OF 2002
```

prec="number"
The number of decimals in the stardate.

<help/>

Provided by module: *Tags: RXML 2 tags*

Gives help texts for tags. If given no arguments, it will list all available tags. By inserting `<help/>` in a page, a full index of the tags available in that particular Roxen WebServer will be presented. If a particular tag is missing from that index, it is not available at that moment. Since all tags are available through modules, that particular tag's module hasn't been added to the Roxen WebServer yet. Ask an administrator to add the module.

Attributes

for="tag"
Gives the help text for that tag.

```
<help for='roxen'/>
<help for='roxen'/>
```

<modified/>

Provided by module: *Tags: RXML 2 tags*

Prints when or by whom a page was last modified, by default the current page.

Attributes

by
Print by whom the page was modified. Takes the same attributes as *<user>*. This attribute requires a user database.

```
This page was last modified by <modified by='1'
 realname='1'/>.
```

date
Print the modification date. Takes all the date attributes in *<date>*.

```
This page was last modified on
<modified date='1' case='lower' type='string'/>.
```

file="path"
Get information about this file rather than the current page.

realfile="path"
Get information from this file in the computers filesystem rather than Roxen Webserver's virtual filesystem.

<number/>

Provided by module: *Tags: RXML 2 tags*

Prints a number as a word.

Attributes

num="number"
Print this number.

```
  <number num='4711'/>
  four thousand seven hundred and eleven
```

language="langcodes"
The language to use.
cat (for catala)
hrv (for croatian)
ces (for czech)
nld (for dutch)
eng (for english)
fin (for finnish)
fra (for french)
deu (for german)
hun (for hungarian)
ita (for italian)
jpn (for japanese)
mri (for maori)
nor (for norwegian)
pol (for polish)
por (for portuguese)
rus (for russian)
srp (for serbian)
slv (for slovenian)
spa (for spanish)
swe (for swedish)

```
  Mitt favoritnummer %x <number num='11' language='
sv'/>.
  Mitt favoritnummer %x elva.
```

```
  Il mio numero preferito Ë <number num='15' langua
ge='it'/>.
  Il mio numero preferito Ë quindici.
```

type="{number, ordered, roman, memory}" (number)
Sets output format.

```
  It was his <number num='15' type='ordered'/
> birthday yesterday.
  It was his 15th birthday yesterday.
```

```
  Only <number num='274589226' type='memory'/
> left on the Internet.
  Only 261.9 Mb left on the Internet.
```

```
  Spock Garfield <number num='17' type='roman'/
> rests here.
  Spock Garfield XVII rests here.
```

<roxen/>

Provided by module: *Tags: RXML 2 tags*

Returns a nice Roxen logo.

Attributes

size="{small, medium, large}" (medium)
Defines the size of the image.

```
  <roxen size='small'/>
<roxen/>
<roxen size='large'/>
  <roxen size='small'/>
<roxen/>
<roxen size='large'/>
```

color="{black, white}" (white)
Defines the color of the image.

```
  <roxen color='black'/>
  <roxen color='black'/>
```

alt="string" ("Powered by Roxen")
The image description.

border="number" (0)
The image border.

class="string"
This cascading style sheet (CSS) definition will be applied on the img element.

target="string"
Names a target frame for the link around the image.
All other attributes will be inherited by the generated img tag.

<user/>

Provided by module: *Tags: RXML 2 tags*

Prints information about the specified user. By default, the full name of the user and her e-mail address will be printed, with a mailto link and link to the home page of that user.
The `<user>` tag requires an authentication module to work.

Attributes

email
Only print the e-mail address of the user, with no link.

```
Email: <user name='foo' email='1'/>
```

link
Include links. Only meaningful together with the realname or email attribute.

name
The login name of the user. If no other attributes are specified, the user's realname and email including links will be inserted.

```
<user name='foo'/>
```

nolink
Don't include the links.

nohomepage
Don't include homepage links.

realname
Only print the full name of the user, with no link.

```
<user name='foo' realname='1'/>
```

Text Tags

Text tags are container tags that process their contents somehow. Examples are *<sort>*, that sorts its contents and *<tablify>*, that creates good looking tables from tab separated text files.

<ai></ai>

Provided by module: *Tags: Indirect href*

Makes it possible to use a database of links. Each link is referred to by a symbolic name instead of the URL.

The database is updated through the configuration interface. The tag is available through the *Indirect href* module.

Attributes

name="string"

Which link to fetch from the database. There is a special case, *name='random'* that will choose a random link from the database.

```
  <ai name='roxen'>Roxen Internet Software</ai>
  <a href="http://
www.roxen.com">Roxen Internet Software</a>
```

<autoformat></autoformat>

Provided by module: *Tags: RXML 2 tags*

Replaces newlines with `<br />`:s'.

```
  <autoformat>
It is almost like
using the pre tag.
</autoformat>
  <br />
It is almost like<br />
using the pre tag.<br />
```

Attributes

p

Replace empty lines with `<p>`:s.

```
  <autoformat p=''>
It is almost like

using the pre tag.
</autoformat>
  <p><br />
It is almost like
</p><p>
using the pre tag.<br />
</p>
```

nobr

Do not replace newlines with `<br />`:s.

nonbsp

Do not turn consecutive spaces into interleaved breakable/nonbreakable spaces. When this attribute is not given, the tag will behave more or less like HTML:s `<pre>` tag, making whitespace indention work, without the usually unwanted effect of really long lines extending the browser window width.

class="string"

This cascading style sheet (CSS) definition will be applied on the p elements.

<case></case>

Provided by module: *Tags: RXML 2 tags*

Alters the case of the contents.

Attributes

case="{upper, lower, capitalize}"

Changes all characters to upper or lower case letters, or capitalizes the first letter in the content.

```
  <case case='upper'>upper</case>
  UPPER
```

```
  <case case='lower'>lower</case>
  lower
```

```
  <case case='capitalize'>capitalize</case>
  Capitalize
```

<charset/> or <charset></charset>

Provided by module: *Tags: RXML 2 tags*

Converts between character sets. The tag can be used both to decode texts encoded in strange character encoding schemas, but also to decide upon the final encoding of the resulting page. All character sets listed in RFC 1345 are supported.

Attributes

in="Character set"

Converts the contents of the charset tag from the character set indicated by this attribute to the internal text representation.

out="Character set"
Sets the output conversion character set of the current request. The page will be sent encoded with the indicated character set.

<comment></comment>

Provided by module: *Tags: RXML 2 tags*

The enclosed text will be removed from the document. The difference from a normal SGML (HTML/XML) comment is that the text is removed from the document, and can not be seen even with *view source* in the browser.

Note that since this is a normal tag, it requires that the content is properly formatted. Therefore it's often better to use the <?comment ... ?> processing instruction tag to comment out arbitrary text (which doesn't contain '?>').

Just like any normal tag, the `<comment>` tag nests inside other `<comment>` tags. E.g:

```
<comment> a <comment> b </comment> c </comment>
```

Here 'c' is not output since the comment starter before 'a' matches the ender after 'c' and not the one before it.

Attributes

preparse
Parse and execute any RXML inside the comment tag. This can be used to do stuff without producing any output in the response. This is a compatibility argument; the recommended way is to use `<nooutput>` instead.

<?comment ?>

Provided by module: *Tags: RXML 2 tags*

Processing instruction tag for comments. This tag is similar to the RXML `<comment>` tag but should be used when commenting arbitrary text that doesn't contain '?>'.

```
<?comment
  This comment will not ever be shown.
?>
```

<default></default>

Provided by module: *Tags: RXML 2 tags*

This tag makes it easier to give default values to "`<select>`" and "`<input>`" form elements. Simply put the `<default>` tag around the form elements to which it should give default values.

This tag is particularly useful in combination with generated forms or forms with generated default values, e.g. by database tags.

Attributes

value="string"
The value or values to set. If several values are given, they are separated with the separator string.

separator="string" (,)
If several values are to be selected, this is the string that separates them.

name="string"
If used, the default tag will only affect form element with this name.

```
<default name='my-select' value='&form.preset;'>
  <select name='my-select'>
    <option value='1'>First</option>
    <option value='2'>Second</option>
    <option value='3'>Third</option>
  </select>
</default>
```

```
<form>
<default value="&form.opt1;,&form.opt2;,&form.opt3;
">
  <input name="opt1" value="yes1" type="checkbox" /
> Option #1
  <input name="opt2" value="yes2" type="checkbox" /
> Option #2
  <input name="opt3" value="yes3" type="checkbox" /
> Option #3
  <input type="submit" />
</default>
</form>
```

<doc></doc>

Provided by module: *Tags: RXML 2 tags*

Eases documentation by replacing "{", "}" and "&" with "<", ">" and "&". No attributes required.

Attributes

quote
Instead of replacing with "{" and "}", "<" and ">" is replaced with "<" and ">".

```
  <doc quote=''>
<table>
 <tr>
    <td> First cell </td>
    <td> Second cell </td>
 </tr>
</table>
</doc>

&lt;table&gt;
 &lt;tr&gt;
    &lt;td&gt; First cell &lt;/td&gt;
    &lt;td&gt; Second cell &lt;/td&gt;
 &lt;/tr&gt;
&lt;/table&gt;
```

pre
The result is encapsulated within a `<pre>` container.

```
  <doc pre=''>
{table}
 {tr}
    {td} First cell {/td}
    {td} Second cell {/td}
 {/tr}
{/table}
</doc>

<pre>
&lt;table&gt;
 &lt;tr&gt;
    &lt;td&gt; First cell &lt;/td&gt;
    &lt;td&gt; Second cell &lt;/td&gt;
```

```
 &lt;/tr&gt;
&lt;/table&gt;
</pre>
```

class="string"
This cascading style sheet (CSS) definition will be applied on the pre element.

<foldlist></foldlist>

Provided by module: *Tags: Folding lists*

This tag is used to build folding lists, that are like `<dl>` lists, but where each element can be unfolded. The tags used to build the lists elements are `<ft>` and `<fd>`.

Attributes

unfolded
Will make all the elements in the list unfolded by default.

foldedsrc
The image to use for folded items. The default is '/internal-roxen-unfold'.

unfoldedsrc
The image to use for unfolded items. The default is '/internal-roxen-fold'.

<ft></ft>

Provided by module: *Tags: Folding lists*

This tag is used within the foldlist tag. The contents of this container, that is not within an fd, tag will be visible both when the element is folded and unfolded.

Attributes

folded
Will make this element folded by default. Overrides an unfolded attribute set in the foldlist tag.

unfolded
Will make this element unfolded by default.

foldedsrc
The image to use for folded items. Overrides the 'foldedsrc' attribute in `<tablist>` for this item.

unfoldedsrc
The image to use for unfolded items. Overrides the 'foldedsrc' attribute in `<tablist>` for this item.

<fd></fd>

Provided by module: *Tags: Folding lists*

The contents of this container will only be visible when the element it is written in is unfolded.

Attributes

```
  <foldlist>
  <ft>
    Heading1
    <fd>Contents 1</fd>
  </ft>
  <ft>
    Heading2
    <fd>Contents 2</fd>
  </ft>
</foldlist>
  <foldlist>
  <ft>
    Heading1
    <fd>Contents 1</fd>
  </ft>
  <ft>
    Heading2
    <fd>Contents 2</fd>
  </ft>
</foldlist>
```

<obox></obox>

Provided by module: *Tags: Outlined box*

This tag creates an outlined box.

Attributes

align="{left, right}"
Vertical alignment of the box.

bgcolor="color"
Color of the background and title label.

fixedleft="number"
Fixed length of line on the left side of the title. The unit is the approximate width of a character.

fixedright="number"
Fixed length of line on the right side of the title. The unit is the approximate width of a character.

left="number"
Length of the line on the left of the title.

outlinecolor="color"
Color of the outline.

outlinewidth="number"
Width, in pixels, of the outline.

right="number"
Length of the line on the right of the title.

spacing="number"
Width, in pixels, of the space in the box.

style="{caption, groupbox}"
Style of the box. Groupbox is default.

textcolor="color"
Color of the text inside the box.

title="string"
Sets the title of the obox.

titlecolor="color"
Color of the title text.

width="number"
Width, in pixels, of the box.
Note that the left and right attributes are constrained by the width argument. If the title is not specified in the argu-

ment list, you can put it in a `<title>` container in the obox contents.

```
  <obox align='left' outlinewidth='5' outlinecolor=
'green' width='200'>
<title>Sample box</title>

This is just a sample box.

</obox>
  <obox align='left' outlinewidth='5' outlinecolor=
'green' width='200'>
<title>Sample box</title>

This is just a sample box.

</obox>
```

<random></random>

Provided by module: *Tags: RXML 2 tags*

Randomly chooses a message from its contents.

Attributes

separator="string"
The separator used to separate the messages, by default newline.

```
  <random separator='#'>Foo#Bar#Baz</random>
  Baz
```

seed="string"
Enables you to use a seed that determines which message to choose.

```
Tip of the day:
<set variable='var.day'><date type='iso' date=''/
></set>
<random seed='var.day'><insert file='tips.txt'/></
random>
```

<replace></replace>

Provided by module: *Tags: RXML 2 tags*

Replaces strings in the contents with other strings.

Attributes

from="string"
String or list of strings that should be replaced.

to="string"
String or list of strings with the replacement strings. Default is the empty string.

separator="string" (,)
Defines what string should separate the strings in the from and to attributes.

type="{word, words}" (word)
Word means that a single string should be replaced. Words that from and to are lists.

<smallcaps></smallcaps>

Provided by module: *Tags: RXML 2 tags*

Prints the contents in smallcaps. If the size attribute is given, font tags will be used, otherwise big and small tags will be used.

```
  <smallcaps>Roxen WebServer</smallcaps>
  <big>R</big><small>OXEN </small><big>W</
big><small>EB</small><big>S</big><small>ERVER</
small>
```

Attributes

space
Put a space between every character.

```
  <smallcaps space=''>Roxen WebServer</smallcaps>
  <big>R </
big><small>O X E N  </
small><big>W </big><small>E B </
small><big>S </
big><small>E R V E R </
small>
```

class="string"
Apply this cascading style sheet (CSS) style on all elements.

smallclass="string"
Apply this cascading style sheet (CSS) style on all small elements.

bigclass="string"
Apply this cascading style sheet (CSS) style on all big elements.

size="number"
Use font tags, and this number as big size.

small="number" (size-1)
Size of the small tags. Only applies when size is specified.

```
  <smallcaps size='6' small='2'>Roxen WebServer</
smallcaps>
  <font size="6">R</font><font size="2">OXEN </
font><font size="6">W</font><font size="2">EB</
font><font size="6">S</font><font size="2">ERVER</
font>
```

<sort></sort>

Provided by module: *Tags: RXML 2 tags*

Sorts the contents.

```
  <sort>Understand!
I
Wee!
Ah,</sort>
  Ah,
I
Understand!
Wee!
```

Attributes

separator="string"
Defines what the strings to be sorted are separated with. The sorted string will be separated by the string.

```
  <sort separator='#'>way?#perhaps#this</sort>
  perhaps#this#way?
```

reverse
Reversed order sort.

```
  <sort reverse=''>backwards?
or
:-)
maybe</sort>
  or
maybe
backwards?
:-)
```

<strlen></strlen>

Provided by module: *Tags: RXML 2 tags*

Returns the length of the contents.

```
  There are <strlen>foo bar gazonk</
strlen> characters
 inside the tag.
  There are 14 characters
 inside the tag.
```

<tablify></tablify>

Provided by module: *Tags: Tablify*

Transforms texts into tables. The default behavior is to use tabs as column delimiters and newlines as row delimiters. The values in the first row as assumed to be the title. Note in the example below how empty rows are ignored. As is shown in the last line a row does not need to be complete for tablify to work properly. These missing cells will get nbsp as content to force all cells to be drawn if borders are on, thus avoiding broken layout when, e.g. a dynamic variable happens to be empty. No attributes are required for tablify to work.

```
  <tablify border='1'>
X Y Z
3 10 77

1 2 10
4 13 3
1 2
</tablify>
  <table face="helvetica,arial" size="2" textcolor=
"#000000" negativecolor="#ff0000" border="1"><tr><t
h align="left">X </
th><th align="left">Y </
th><th align="left">Z </th></tr>
<tr><td align="left">3</td><td align="left">10</
td><td align="left">77</td></tr>
<tr><td align="left">1</td><td align="left">2</
td><td align="left">10</td></tr>
<tr><td align="left">4</td><td align="left">13</
td><td align="left">3</td></tr>
<tr><td align="left">1</td><td align="left">2</
td><td colspan="1"> </td></tr>
</table>
```

Tablify also prescans the entire table to find the widest number of cells.

```
  <tablify border='1'>
A B
a b
aa bb
aaa bbb ops!
</tablify>
  <table face="helvetica,arial" size="2" textcolor=
"#000000" negativecolor="#ff0000" border="1"><tr><t
h align="left">A </
th><th align="left">B </
th><td colspan="1"> </td></tr>
<tr><td align="left">a</td><td align="left">b</
td><td colspan="1"> </td></tr>
<tr><td align="left">aa</td><td align="left">bb</
td><td colspan="1"> </td></tr>
<tr><td align="left">aaa</td><td align="left">bbb</
td><td align="left">ops!</td></tr>
</table>
```

Attributes

rowseparator="string" (newline)
Defines the character or string used to seperate the rows.

cellseparator="string" (tab)
Defines the character or string used to seperate the cells.

```
  <tablify cellseparator=',>
Element, Mass
H, 1.00797
He, 4.0026
Li, 6.939
</tablify>
  <table face="helvetica,arial" size="2" textcolor=
"#000000" negativecolor="#ff0000"><tr><th align="le
ft">Element </
th><th align="left"> Mass </th></tr>
<tr><td align="left">H</
td><td align="left"> 1.00797</td></tr>
<tr><td align="left">He</
td><td align="left"> 4.0026</td></tr>
<tr><td align="left">Li</
td><td align="left"> 6.939</td></tr>
</table>
```

intable
If the intable attribute is set the tablify module will parse the indata as a table.

```
  <tablify nice='1' intable='1'>
<table><tr><th>Element</th><th>Mass</th></tr>
<tr><td>H</td><td>1.00797</td></tr>
<tr><td>He</td><td>4.0026</td></tr>
<tr><td>Li</td><td>6.939</td></tr>
</table>
</tablify>
  <table bgcolor="#000000" border="0" cellspacing="
0" cellpadding="1">
<tr><td>
<table border="0" cellspacing="0" cellpadding="4" w
idth="100%">
```

```
<tr bgcolor="#112266">
<th align="left"><font color="#ffffff">Element</
font> </
th><th align="left"><font color="#ffffff">Mass</
font> </
th><th align="left"><font color="#ffffff">></
font> </th></tr>
<tr bgcolor="#ffffff"><td align="left">H  
;</td><td align="left">1.00797  </
td><td align="left">  </td></tr>
<tr bgcolor="#ddeeff"><td align="left">He &nbs
p;</td><td align="left">4.0026  </
td><td align="left">  </td></tr>
<tr bgcolor="#ffffff"><td align="left">Li &nbs
p;</td><td align="left">6.939  </
td><td align="left">  </td></tr>
</table></td></tr>
</table>
```

notitle

Don't add a title to the columns and treat the first row in the indata as real data instead.

interactive-sort

Makes it possible for the user to sort the table with respect to any column.

sortcol="number"

Defines which column to sort the table with respect to. The leftmost column is number 1. Negative value indicate reverse sort order.

min="number"

Decides which of the input rows should be the first one to be displayed. This can be used to skip unwanted rows in the beginning of the data. The first row after the heading is row number 1.

max="number"

Decides which of the input rows should be the last one to be displayed. This can be used to limit the the output to a maximum number of rows.

```
  <tablify min='2' max='4'>
Stuff
one
two
three
four
five
six
</tablify>
  <table face="helvetica,arial" size="2" textcolor=
"#000000" negativecolor="#ff0000"><tr><th align="le
ft">Stuff </th></tr>
<tr><td align="left">two</td></tr>
<tr><td align="left">three</td></tr>
<tr><td align="left">four</td></tr>
</table>
```

negativecolor="color" (#ff0000)

The color of negative values in economic fields.

border="number"

Defines the width of the border. Default is 2 in nice and nicer modes. Otherwise undefined. The value is propagated into the resulting table tag if neither nice nor nicer is used.

cellspacing="number"

Defines the cellspacing attribute. Default is 0 in nice and nicer modes. Otherwise undefined. The value is propagated into the resulting table tag if neither nice nor nicer is used.

cellpadding="number"

Defines the cellpadding attribute. Default is 4 in nice and nicer modes. Otherwise undefined. The value is propagated into the resulting table tag if neither nice nor nicer is used.

width="number"

Defines the width of the table.

cellalign="{left, center, right}"

Defines how the cell contents should be align by default. The value is propagated into the resulting td tags if neither nice nor nicer is used.

cellvalign="{top, middle, bottom}"

Defines how the cell contents should be verically aligned. The value is propagated into the resulting td tags if neither nice nor nicer is used.

The 'nice' attribute

nice

Add some extra layout to the table. More specifically it creates a bakcground table with another color and then colors all the cells in the inner table. All attributes below only applies in nice or nicer mode.

```
  <tablify nice='1' cellseparator=','>
Element, Mass
H, 1.00797
He, 4.0026
Li, 6.939
</tablify>
  <table bgcolor="#000000" border="0" cellspacing="
0" cellpadding="1">
<tr><td>
<table border="0" cellspacing="0" cellpadding="4" w
idth="100%">
<tr bgcolor="#112266">
<th align="left">Element</
font> </
th><th align="left"> Mass</
font> </th></tr>
<tr bgcolor="#ffffff"><td align="left">H  
;</td><td align="left"> 1.00797  </td></
tr>
<tr bgcolor="#ddeeff"><td align="left">He &nbs
p;</td><td align="left"> 4.0026  </td></
tr>
<tr bgcolor="#ffffff"><td align="left">Li &nbs
p;</td><td align="left"> 6.939  </td></
tr>
</table></td></tr>
</table>
```

```
  <tablify nice='1' cellseparator=',' cellspacing='
1'>
Element, Mass
H, 1.00797
He, 4.0026
Li, 6.939
</tablify>
  <table bgcolor="#000000" border="0" cellspacing="
0" cellpadding="1">
<tr><td>
<table border="0" cellspacing="1" cellpadding="4" w
idth="100%">
<tr bgcolor="#112266">
<th align="left">Element</
```

```
font> </
th><th align="left"><font color="#ffffff"> Mass</
font> </th></tr>
<tr bgcolor="#ffffff"><td align="left">H  
;</td><td align="left"> 1.00797  </td></
tr>
<tr bgcolor="#ddeeff"><td align="left">He &nbs
p;</td><td align="left"> 4.0026  </td></
tr>
<tr bgcolor="#ffffff"><td align="left">Li &nbs
p;</td><td align="left"> 6.939  </td></
tr>
</table></td></tr>
</table>
```

grid="number"
Draws a grid with the thickness given.

```
  <tablify nice='1' grid='1'>
Element Mass
H 1.00797
He 4.0026
Li 6.939
</tablify>
  <table bgcolor="#000000" border="0" cellspacing="
0" cellpadding="1">
<tr><td>
<table border="1" cellspacing="0" cellpadding="4" w
idth="100%">
<tr bgcolor="#112266">
<th align="left"><font color="#ffffff">Element</
font> </
th><th align="left"><font color="#ffffff">Mass</
font> </th></tr>
<tr bgcolor="#ffffff"><td align="left">H  
;</td><td align="left">1.00797  </td></
tr>
<tr bgcolor="#ddeeff"><td align="left">He &nbs
p;</td><td align="left">4.0026  </td></
tr>
<tr bgcolor="#ffffff"><td align="left">Li &nbs
p;</td><td align="left">6.939  </td></tr>
</table></td></tr>
</table>
```

bordercolor="color" (#000000)
The color of the border.

titlebgcolor="color" (#112266)
The background color of the title.

titlecolor="color" (#ffffff)
The color of the title.

modulo="number"
Defines how many rows in a row should have the same color.

```
  <tablify nice='1' cellseparator=',' modulo='2'>
Element, Mass
H, 1.00797
He, 4.0026
Li, 6.939
Be, 9.0122
B, 10.811
</tablify>
  <table bgcolor="#000000" border="0" cellspacing="
0" cellpadding="1">
<tr><td>
<table border="0" cellspacing="0" cellpadding="4" w
idth="100%">
<tr bgcolor="#112266">
<th align="left"><font color="#ffffff">Element</
font> </
th><th align="left"><font color="#ffffff"> Mass</
font> </th></tr>
<tr bgcolor="#ffffff"><td align="left">H  
;</td><td align="left"> 1.00797  </td></
tr>
<tr bgcolor="#ffffff"><td align="left">He &nbs
p;</td><td align="left"> 4.0026  </td></
tr>
<tr bgcolor="#ddeeff"><td align="left">Li &nbs
p;</td><td align="left"> 6.939  </td></
tr>
<tr bgcolor="#ddeeff"><td align="left">Be &nbs
p;</td><td align="left"> 9.0122  </td></
tr>
<tr bgcolor="#ffffff"><td align="left">B  
;</td><td align="left"> 10.811  </td></
tr>
</table></td></tr>
</table>
```

oddbgcolor="color" (#ffffff)
The first background color.

evenbgcolor="color" (#ddeeff)
The second background color.

The 'nicer' attribute

nicer
Add some extra extra layout to the table. Compared with nice-mode it gtexts the column titles and adds a font-tag in all cells. All attributes below only applies in nicer mode. Nicer requires the gtext module.

```
  <tablify nicer='1'>
Element Mass
H 1.00797
He 4.0026
Li 6.939
</tablify>
  <tablify nicer='1'>
Element Mass
H 1.00797
He 4.0026
Li 6.939
</tablify>
```

noxml
Don't terminate the images with slashes, as required by XML.

font="text" (lucida)
The font gtext should use to write the column titles with.

```
  <tablify nicer='1' cellseparator=', ' font='andov
er' fontsize='24'>
Element, Mass
H, 1.00797
He, 4.0026
Li, 6.939
</tablify>
  <tablify nicer='1' cellseparator=', ' font='andov
er' fontsize='24'>
Element, Mass
H, 1.00797
He, 4.0026
Li, 6.939
</tablify>
```

fontsize="int" (13)
The size of the gtext font used to write the column titles with.

scale="float"
Scales the gtext font used to write the column titles with.

textcolor="color" (#000000)
The color of the text. This will also work with economic fields in any mode.

size="number" (2)
The size of the table text.

face="string" (helvetica,arial)
The font of the table text, e.g. the value of the face attribute in the font tag that encloses every cell.

<fields></fields>

Provided by module: *Tags: Tablify*

The container 'fields' may be used inside the tablify container to describe the type of contents the fields in a column has. Available fields are:

- text (default)
- left
- center
- right
- num
- int
- economic-int
- float
- economic-float

All fields except text overrides the cellvalign attribute.

```
  <tablify nice='nice'>
<fields separator=','>
text,center,right,float,economic-
float,int,economic-int
</fields>
text center right float economic-float int economic-
int
123.14 123.14 123.14 123.14 123.14 123.14 123.14
56.8 56.8 56.8 56.8 56.8 56.8 56.8
-2 -2 -2 -2 -2 -2 -2
</tablify>
  <table bgcolor="#000000" border="0" cellspacing="
0" cellpadding="1">
<tr><td>
<table border="0" cellspacing="0" cellpadding="4" w
idth="100%">
<tr bgcolor="#112266">
<th align="left"><font color="#ffffff">text</
font> </
th><th align="left"><font color="#ffffff">center</
font> </
th><th align="left"><font color="#ffffff">right</
font> </
th><th align="left"><font color="#ffffff">float</
font> </
th><th align="left"><font color="#ffffff">economic-
float</font> </
th><th align="left"><font color="#ffffff">int</
font> </
th><th align="left"><font color="#ffffff">economic-
int</font> </th></tr>
<tr bgcolor="#ffffff"><td align="left">123.14 
 </td><td align="center">123.14  </
td><td align="right">123.14  </
td><td align="right"><table border="0" cellpadding=
"0" cellspacing="0"><tr><td align="right">123</
td><td>.</td><td align="left" width="30">14</td></
tr></table></
td><td align="right"><table border="0" cellpadding=
"0" cellspacing="0"><tr><td align="right"><font col
or="#000000">123</font></
td><td><font color="#000000">.</font></
td><td align="left" width="30"><font color="#000000
">14</font></td></tr></table></
td><td align="right">123</
td><td align="right"><font color="#000000">123</
font></td></tr>
<tr bgcolor="#ddeeff"><td align="left">56.8 &n
bsp;</td><td align="center">56.8  </
td><td align="right">56.8  </
td><td align="right"><table border="0" cellpadding=
"0" cellspacing="0"><tr><td align="right">56</
td><td>.</td><td align="left" width="30">8</td></
tr></table></
td><td align="right"><table border="0" cellpadding=
"0" cellspacing="0"><tr><td align="right"><font col
or="#000000">56</font></
td><td><font color="#000000">.</font></
td><td align="left" width="30"><font color="#000000
">8</font></td></tr></table></
td><td align="right">57</
td><td align="right"><font color="#000000">57</
font></td></tr>
<tr bgcolor="#ffffff"><td align="left">-
2  </td><td align="center">-
2  </td><td align="right">-
2  </
td><td align="right"><table border="0" cellpadding=
"0" cellspacing="0"><tr><td align="right">-2</
td><td>.</td><td align="left" width="30">0</td></
tr></table></
td><td align="right"><table border="0" cellpadding=
"0" cellspacing="0"><tr><td align="right">-2</
td><td><font color="#ff0000">.</font></
td><td align="left" width="30">0</td></tr></table></
td><td align="right">-2</
td><td align="right">-2</
td></tr>
</table></td></tr>
</table>
```

Attributes

separator="string"
Defines the field type separator.
The fields types are separated by

1. The value given in the separator attribute to fields.
2. The value given in the cellseparator attribute to tablify.
3. Tab.

<trimlines></trimlines>

Provided by module: *Tags: RXML 2 tags*

Removes all empty lines from the contents.

```
  <pre><trimlines>
See how all this junk
```

```
just got zapped?

</trimlines></pre>
  <pre>See how all this junk
just got zapped?</pre>
```

<wash-html></wash-html>

Provided by module: *Tags: HTML washer*

This tag is mostly useful for turning user freetext input from a form into HTML intelligently, by turning sections of the text separated by more than one newline into paragraphs , filtering out or explicitly allowing some HTML tags in the input and creating <a>anchor-links out of potential www-addresses.

Attributes

keep-all

Leave all tags containing info intact. Overrides the value of keep-tags and keep-containers. This attribute is useful together with the attributes *unparagraphify* and *unlink*.

```
  <wash-html keep-all='1'>
  Some text, <i>italic</i>, <b>bold</b>,
  <i><b>bold italic</b></i>.

  <hr />A little image:<img src='/internal-roxen-
next' />.
</wash-html>
  <wash-html keep-all='1'>
  Some text, <i>italic</i>, <b>bold</b>,
  <i><b>bold italic</b></i>.

  <hr />A little image:<img src='/internal-roxen-
next' />.
</wash-html>
```

keep-tags="list"

Comma-separated array of empty element <tags> not to filter. Quote all other empty element tags, i.e. transform "<", ">" and "&" to "<", ">" and "&".

```
  <wash-html keep-tags='hr'>
  Some text, <i>italic</i>, <b>bold</b>,
  <i><b>bold italic</b></i>.

  <hr />A litle image:<img src='/internal-roxen-
next' />.
</wash-html>

  Some text, &lt;i&gt;italic&lt;/
i&gt;, &lt;b&gt;bold&lt;/b&gt;,
  &lt;i&gt;&lt;b&gt;bold italic&lt;/b&gt;&lt;/
i&gt;.

  <hr>A litle image:&lt;img src='/internal-roxen-
next' /&gt;.
```

keep-containers="list"

Comma-separated array of <container>...</> tags not to filter. Quote all other container tags, i.e. transform "<", ">" and "&" to "<", ">" and "&".

```
  <wash-html keep-containers='b'>
  Some text, <i>italic</i>, <b>bold</b>,
  <i><b>bold italic</b></i>.

  <hr />A little image:<img src='/internal-roxen-
next' />.
</wash-html>

  Some text, &lt;i&gt;italic&lt;/i&gt;, <b>bold</
b>,
  &lt;i&gt;<b>bold italic</b>&lt;/i&gt;.

  &lt;hr /&gt;A little image:&lt;img src='/
internal-roxen-next' /&gt;.
```

linkify

Makes text that looks like it might be useful as a link, e g http://www.roxen.com/, into a link. Text that starts with "http://", "https://", "ftp://", "www." or "http." will be converted to a clickable link with the text as the link label.

```
  <wash-html linkify='a' keep-containers='a' keep-
tags='br'>
  <a href="http://docs.roxen.com">Roxen docs</
a><br />
  http://pike.roxen.com<br />
  www.roxen.com
</wash-html>

  <a href="http://docs.roxen.com">Roxen docs</
a><br>
  <a href='http://pike.roxen.com'>http://
pike.roxen.com</a><br>
  <a href='http://www.roxen.com'>http://
www.roxen.com</a>
```

unlinkify

Undo a linkify-conversion. Only the links that has the same label as address will be converted to plain text.

```
  <wash-html unlinkify='1' keep-tags='br' keep-
containers='a'>
  <a href="http://www.roxen.com">http://
www.roxen.com</a><br />
  <a href="http://www.roxen.com">Roxen IS</a>
</wash-html>

  http://www.roxen.com<br>
  <a href="http://www.roxen.com">Roxen IS</a>
```

paragraphify

If more than one newline exists between two text elements, this attribute automatically makes the next text element into a paragraph.

```
  <wash-html paragraphify='1'>
A Paragraph

Another paragraph.
Some more text to the same paragraph.
</wash-html>
  &lt;p&gt;
A Paragraph&lt;/p&gt;
&lt;p&gt;Another paragraph.
Some more text to the same paragraph.
&lt;/p&gt;
```

unparagraphify

Turn paragraph breaks into double newlines instead.

```
  <wash-html unparagraphify='1'>
<p>A Paragraph</p>
<p>Another paragraph.
Some more text to the same paragraph.</p>
</wash-html>

A Paragraph

Another paragraph.
Some more text to the same paragraph.
```

The `<pre>` is only used in the example for layout-purposes.

close-tags

Terminate all tags with an ending slash, making them XML-compliant.

Variable Tags

Variable tags are mostly used to create new variables or change a variable's contents in some manner. Some of the tags can also be used to create new tags or insert various contents of variables, files, etc into a web page.

Variables are mostly created with tags like *<set>* and *<define>* and are reached through entities (scope.variable). *[More about variables]*

<append/> or <append></append>

Provided by module: *Tags: RXML 2 tags*

Appends a value to a variable. The variable attribute and one more is required.

Attributes

variable="string"
The name of the variable.

value="string"
The value the variable should have appended.

```
   <set variable='var.ris' value='Roxen'/>
 <append variable='var.ris' value=' Internet Softwa
re'/>
 &var.ris;
```

```
 Roxen Internet Software
```

from="string"
The name of another variable that the value should be copied from.

<copy-scope/>

Provided by module: *Tags: RXML 2 tags*

Copies the content of one scope into another scope

Attributes

from="scope name"
The name of the scope the variables are copied from.

to="scope name"
The name of the scope the variables are copied to.

<dec/>

Provided by module: *Tags: RXML 2 tags*

Subtracts 1 from a variable.

Attributes

variable="string"
The variable to be decremented.

value="number" (1)
The value to be subtracted.

<define></define>

Provided by module: *Tags: RXML 2 tags*

Defines variables, tags, containers and if-callers.

The values of the attributes given to the defined tag are available in the scope created within the define tag.

Attributes

variable="name"
Sets the value of the variable to the contents of the container.

tag="name"
Defines a tag that outputs the contents of the container.

```
  <define tag="hi">Hello &_.name;!</define>
<hi name="Martin"/>
```

```
Hello Martin!
```

container="name"
Defines a container that outputs the contents of the container.

if="name"
Defines an if-caller that compares something with the contents of the container.

trimwhites
Trim all white space characters from the beginning and the end of the contents.

preparse="preparse"
Sends the definition through the RXML parser when defining. (Without this attribute, the definition is only RXML parsed when it is invoked.)

&_.args;
Provided by *Tags: RXML 2 tags*
The full list of the attributes, and their arguments, given to the tag.

&_.contents;
Provided by *Tags: RXML 2 tags*
The containers contents.

&_.rest-args;
Provided by *Tags: RXML 2 tags*
A list of the attributes, and their arguments, given to the tag, excluding attributes with default values defined.

<attrib></attrib>

Provided by module: *Tags: RXML 2 tags*

When defining a tag or a container the tag `<attrib>` can be used to define default values of the attributes that the tag/container can have. The attrib tag must be the first tag(s) in the define tag.

Attributes

name="name"
The name of the attribute which default value is to be set.

<contents/>

Provided by module: *Tags: RXML 2 tags*

As the contents entity, but unquoted.

<inc/>

Provided by module: *Tags: RXML 2 tags*

Adds 1 to a variable.

Attributes

variable="string"
The variable to be incremented.

value="number" (1)
The value to be added.

<insert/>

Provided by module: *Tags: RXML 2 tags*

Inserts a file, variable or other object into a webpage.

Attributes

quote="{html, none}"
How the inserted data should be quoted. Default is "html", except for the file plugin where it's "none".

<insert file/>

Provided by module: *Tags: RXML 2 tags*

Inserts the contents of a file. It reads files in a way similar to if you fetched the file with a browser, so the file may be parsed before it is inserted, depending on settings in the RXML parser. Most notably which kinds of files (extensions) that should be parsed. Since it reads files like a normal request, e.g. generated pages from location modules can be inserted. Put the tag *<eval>* around `<insert>` if the file should be parsed after it is inserted in the page. This enables RXML defines and scope variables to be set in the including file (as opposed to the included file). You can also configure the file system module so that files with a certain extension can not be downloaded, but still inserted into other documents.

Attributes

file="string"
The virtual path to the file to be inserted.

```
<eval><insert file='html_header.inc'/></eval>
```

<insert href/>

Provided by module: *Tags: Additional RXML tags*

Inserts the contents at that URL. This function has to be enabled in the *Additional RXML tags* module in the Roxen WebServer configuration interface. The page download will block the current thread, and if running unthreaded, the whole server. There is no timeout in the download, so if the server connected to hangs during transaction, so will the current thread in this server.

Attributes

href="string"
The URL to the page that should be inserted.

nocache="string"
If provided the resulting page will get a zero cache time in the RAM cache. The default time is up to 60 seconds depending on the cache limit imposed by other RXML tags on the same page.

<insert realfile/>

Provided by module: *Tags: RXML 2 tags*

Inserts a raw, unparsed file. The disadvantage with the realfile plugin compared to the file plugin is that the realfile plugin needs the inserted file to exist, and can't fetch files from e.g. an arbitrary location module. Note that the realfile insert plugin can not fetch files from outside the virtual file system.

Attributes

realfile="string"
The virtual path to the file to be inserted.

<insert scopes/>

Provided by module: *Tags: RXML 2 tags*

Inserts a listing of all present variable scopes.

Attributes

scopes="{full, plain}"
Sets how the output should be formatted.

```
  <insert scopes='plain'/>
  client, cookie, form, modvar, page, roxen, ssi, u
ser and var
```

<insert variable/>

Provided by module: *Tags: RXML 2 tags*

Inserts the value of a variable.

Attributes

variable="string"
The name of the variable.

scope="string"
The name of the scope, unless given in the variable attribute.

index="number"
If the value of the variable is an array, the element with this index number will be inserted. 1 is the first element. -1 is the last element.

split="string"
A string with which the variable value should be splitted into an array, so that the index attribute may be used.

<insert variables/>

Provided by module: *Tags: RXML 2 tags*

Inserts a listing of all variables in a scope.It is possible to create a scope with an infinite number of variables set. In this case the programmer of that scope decides which variables that should be listable, i.e. this will not cause any problem except that all variables will not be listed. It is also possible to hide variables so that they are not listed with this tag.

Attributes

variables="{full, plain}"
Sets how the output should be formatted.

```
  <pre>
<insert variables='full' scope='roxen'/>
</pre>
  <pre>
domain=&#34;megalon.roxen.com&#34;
hits=0
hits-per-minute=0
locale=&#34;eng&#34;
path=0
pike-version=&#34;Pike v7.2 release 360&#34;
sent=0
sent-kbit-per-second=&#34;0.00&#34;
sent-mb=&#34;0.00&#34;
sent-per-minute=0
server=&#34;http://megalon.roxen.com:21033/&#34;
ssl-strength=168
time=1026491872
unique-
id=&#34;904d68ed9d371ccc3f132e980ca029b1&#34;
uptime=146
uptime-days=0
uptime-hours=0
uptime-minutes=2
version=&#34;Roxen/2.2.295&#34;
</pre>
```

scope
The name of the scope that should be listed, if not the present scope.

<scope></scope>

Provided by module: *Tags: RXML 2 tags*

Creates a new variable scope. Variable changes inside the scope container will not affect variables in the rest of the page.

Attributes

extend="name" (form)
If set, all variables in the selected scope will be copied into the new scope. NOTE: if the source scope is "magic", as e.g. the roxen scope, the scope will not be copied, but rather linked and will behave as the original scope. It can be useful to create an alias or just for the convinience of refering to the scope as "_".

scope="name" (form)
The name of the new scope, besides "_".

<set/> or <set></set>

Provided by module: *Tags: RXML 2 tags*

Sets a variable in any scope that isn't read-only.

```
<set variable='var.language'>Pike</set>
```

Attributes

variable="string"
The name of the variable.

```
<set variable='var.foo' value='bar'/>
```

value="string"
The value the variable should have.

expr="string"
An expression whose evaluated value the variable should have.

from="string"
The name of another variable that the value should be copied from.

split="string"
The value will be splitted by this string into an array.
If none of the above attributes are specified, the variable is unset. If debug is currently on, more specific debug information is provided if the operation failed. See also: <*append*> and <*debug*>.

<sprintf></sprintf>

Provided by module: *Tags: Additional RXML tags*

Prints out variables with the formating functions availble in the Pike function sprintf. Refer to the Pike reference manual for a complete description.

Attributes

format="string"
The formatting string.

split="charater"

If used, the tag content will be splitted with the given string.

```
  <sprintf format='#%02x%02x%02x' split=',>250,0,3
3</sprintf>
  #fa0021
```

<sscanf></sscanf>

Provided by module: *Tags: Additional RXML tags*

Extract parts of a string and put them in other variables. Refer to the sscanf function in the Pike reference manual for a complete description.

Attributes

variables="list"
A comma separated list with the name of the variables that should be set.

```
  <sscanf variables='form.year,var.month,var.day'
format='%4d%2d%2d'>19771003</sscanf>
&form.year;-&var.month;-&var.day;

1977-10-3
```

scope="name"
The name of the fallback scope to be used when no scope is given.

```
  <sscanf variables='year,month,day' scope='var'
 format='%4d%2d%2d'>19801228</sscanf>
&var.year;-&var.month;-&var.day;<br />
<sscanf variables='form.year,var.month,var.day'
 format='%4d%2d%2d'>19801228</sscanf>
&form.year;-&var.month;-&var.day;

1980-12-28<br />

1980-12-28
```

return="name"
If used, the number of successfull variable 'extractions' will be available in the given variable.

<undefine/>

Provided by module: *Tags: RXML 2 tags*

Removes a definition made by the define container. One attribute is required.

Attributes

variable="name"
Undefines this variable.

```
    <define variable='var.hepp'>hopp</define>
  &var.hepp;
  <undefine variable='var.hepp'/>
  &var.hepp;

  hopp
```

tag="name"
Undefines this tag.

container="name"
Undefines this container.

if="name"
Undefines this if-plugin.

<unset/>

Provided by module: *Tags: RXML 2 tags*

Unsets a variable, i.e. removes it.

Attributes

variable="string"
The name of the variable.

```
  <set variable='var.jump' value='do it'/>
&var.jump;
<unset variable='var.jump'/>
&var.jump;

do it
```

<use></use>

Provided by module: *Tags: RXML 2 tags*

Reads *tag definitions*, user defined *if plugins* and *variables* from a file or package and includes into the current page.

Attributes

packageinfo
Show a list of all available packages.

package="name"
Reads all tags, container tags and defines from the given package. Packages are files located by default in *../rxml_packages/*.

file="path"
Reads all tags and container tags and defines from the file.
This file will be fetched just as if someone had tried to fetch it with an HTTP request. This makes it possible to use Pike script results and other dynamic documents. Note, however, that the results of the parsing are heavily cached for performance reasons. If you do not want this cache, use `<insert file='...' nocache='1'>` instead.

info
Show a list of all defined tags/containers and if arguments in the file.

<vform></vform>

Provided by module: *Tags: Verified form*

Creates a self verifying form. You can use all standard HTML-input widgets in this container as well.

```
<vform>
  <vinput name='mail' type='email'>&_.warning;</
vinput>
  <input type='hidden' name='user' value='&form.use
rid;' />
  <input type='submit' />
</vform>
<then><redirect to='other_page.html' /></then>
<else>No, this form is still not valid</else>
```

Attributes

hide-if-verified
Hides the form if it is verified

<clear/>

Provided by module: *Tags: Verified form*

Resets all the widgets to their initial values.

Attributes

value="string"
The text in the button.

<if vform-failed></if>

Provided by module: *Tags: Verified form*

If used with empty argument this will be true if the complete form is failed, otherwise only if the named field failed.

<if vform-verified></if>

Provided by module: *Tags: Verified form*

If used with empty arguemnt this will be true if the complete form so far is verified, otherwise only if the named field was successfully verified.

<reload/>

Provided by module: *Tags: Verified form*

Reload the page without variable checking.

Attributes

value="string"
The text on the button.

<verify-fail/>

Provided by module: *Tags: Verified form*

If put in a vform tag, the vform will always fail.This is useful e.g. if you put the verify-fail tag in an if tag.

<verify-ok/>

Provided by module: *Tags: Verified form*

If put in a vform tag, the vform will always be verified.This tag is probably only useful when the name-attribute inside the tag is set. If it is it will force that specific vform-variable as verified ok even if the <vinput>-tag that tested the variable failed.

Attributes

name="string"
The name of the vform variable to force as verified ok.

<vinput></vinput>

Provided by module: *Tags: Verified form*

Creates a self verifying input widget.

Attributes

fail-if-failed="name"
The verification of this variable will always fail if the verification of a named variable also failed.

ignore-if-false
Don't verify if the false flag i set.

ignore-if-failed="name"
Don't verify if the verification of a named variable failed.

ignore-if-verified="name"
Don't verify if the verification of a named variable succeeded.

name="string"
The name of the variable that should be set.

value="anything"
The default value of this input widget.

scope="name" (vinput)
The name of the scope that is created in this tag.

trim
Trim the variable before verification.

type="{int, float, email, date, text, string, password}"
Set the type of the data that should be input, and hence what widget should be used and how the input should be verified.

minlength="number"
Verify that the variable has at least this many characters. Only available when using the type password, string or text.

maxlength="number"
Verify that the variable has at most this many characters. Only available when using the type password, string or text.

is="empty"
Verify that the variable is empty. Pretty useless... Only available when using the type password, string or text.

glob="pattern"

Verify that the variable match a certain glob pattern. Only available when using the type password, string or text.

regexp="pattern"
Verify that the variable match a certain regexp pattern. Only available when using the type password, string or text.

case="{upper, lower}"
Verify that the variable is all uppercased (or all lowercased). Only available when using the type password, string or text.

equal="string"
Verify that the variable is equal to a given string. Pretty useless... Only available when using the type password, string or text.

disable-domain-check
Only available when using the email type. When set the email domain will not be checked against a DNS to verify that it does exists.

mode="{before, after, complex}"
Select how to treat the contents of the vinput container. *Before* puts the contents before the input tag, and *after* puts it after, in the event of failed verification. If complex, use one tag `<verified>` for what should be outputted in the event of successful verification tag `<failed>` for every other event.

```
<table>
<tr><td>upper</td>
<vinput name='a' case='upper' mode='complex'>
<verified><td bgcolor='green'>&_.input:none;</td></
verified>
<failed><td bgcolor='red'>&_.input:none;</td></
failed>
</vinput></tr>
<tr><td><input type='submit' /></td></tr>
</table>
```

min="number"
Check that the number is at least the given. Only available when using the type int or float.

max="number"
Check that the number is not greater than the given. Only available when using the type int or float.

optional
Indicates that the variable should only be tested if it does contain something.

&_.input;
Provided by *Tags: Verified form*
The input tag, in complex mode.

&_.warning;
Provided by *Tags: Verified form*
May contain a explaination of why the test failed.

<failed></failed>

Provided by module: *Tags: Verified form*

The content will only be shown if the variable failed to verify, in complex mode.

<verified></verified>

Provided by module: *Tags: Verified form*

The content will only be shown if the variable was verfied, in complex mode.

Protocol Tags

Protocol tags are tags that somehow use or manipulate various protocol features on a lower level, e.g. they can modify special features of the HTTP protocol, alter the URL or send messages on other protocols than HTTP.

<aconf></aconf>

Provided by module: *Tags: RXML 2 tags*

Creates a link that can modify the config states in the cookie RoxenConfig. In practice it will add <keyword>/ right after the server in the URL. E.g. if you want to remove the config state bacon and add config state egg the first "directory" in the path will be <-bacon,egg>. If the user follows this link the WebServer will understand how the RoxenConfig cookie should be modified and will send a new cookie along with a redirect to the given url, but with the first "directory" removed. The presence of a certain config state can be detected by the <*if config*> tag.

Attributes

href="uri"
Indicates which page should be linked to, if any other than the present one.

add="string"
The config state, or config states that should be added, in a comma separated list.

drop="string"
The config state, or config states that should be dropped, in a comma separated list.

class="string"
This cascading style sheet (CSS) class definition will apply to the a-element.
All other attributes will be inherited by the generated `<a>` tag.

<apre></apre>

Provided by module: *Tags: RXML 2 tags*

Creates a link that can modify prestates. Prestates can be seen as valueless cookies or toggles that are easily modified by the user. The prestates are added to the URL. If you set the prestate "no-images" on "http://www.demolabs.com/index.html" the URL would be "http://www.demolabs.com/(no-images)/". Use <*if prestate*> to test for the presence of a prestate. `<apre>` works just like the `<a href='...'>` container, but if no "href" attribute is specified, the current page is used.

Attributes

href="uri"
Indicates which page should be linked to, if any other than the present one.

add="string"
The prestate or prestates that should be added, in a comma separated list.

drop="string"
The prestate or prestates that should be dropped, in a comma separated list.

class="string"
This cascading style sheet (CSS) class definition will apply to the a-element.

<auth-required/>

Provided by module: *Tags: RXML 2 tags*

Adds an HTTP auth required header and return code (401), that will force the user to supply a login name and password. This tag is needed when using access control in RXML in order for the user to be prompted to login.

Attributes

realm="string" (document access)
The realm you are logging on to, i.e "Demolabs Intranet".

message="string"
Returns a message if a login failed or cancelled.

<email></email>

Provided by module: *Tags: E-mail module*

The `<email>` sends MIME compliant mail to a mail server using the (E)SMTP protocol. The content is sent as raw text Attribute default values can be changed within the *E-mail module's* administration interface.

Attributes

server="URL" (localhost)
The hostname of the machine that operates the mail server.

subject="" ("[* No Subject *]")
The subject line.

from="" ((empty))
The email address of sender.

to="" ((empty))
The list of recipient email address(es). Separator character can be defined by the 'separator' attribute.

separator="" (,)

The separator character for the recipient list.

mimetype="MIME type"
Overrides the MIME type of the body.

charset="" (iso-8859-1)
The charset of the body.

```
<email from="foo@bar.com" to="johny@pub.com|pity@bu
fet.com|ely@rest.com"
separator="|" charset="iso-8859-
2" server="mailhub.anywhere.org" >
 This is the contents.
</email>
```

<attachment/> or <attachment></attachment>

Provided by module: *Tags: E-mail module*

This tag/subcontainer is designed for adding attachments to the mail.

There are two different kinds of attachments; file and inline. File attachments require the *file* attribute while inline attachments are written inline. Inline attachments can for instance be a text or a binary (e.g. output from a database).

```
<email from="foo@bar.com" to="johny@pub.com|pity@bu
fet.com|ely@rest.com"
separator="|" charset="iso-8859-
2" server="mailhub.anywhere.org">

<header name="X-foo-header" value="one two three" /
>
<header name="Importance">Normal</header>
<header name="X-MSMail-Priority" value="Normal" />
 This is the contents.

<attachment file="/images/hello.gif" />
<attachment file="/excel/
p_123.xls" name="partners.xls" />

<attachment name="partners.txt" >
      company1        1.2345  abc
      company2        2.345   ix
      company8        3.4567  az
</attachment>
</email>
```

Attributes

file="path"
The path to the file in the virtual filesystem. If this attribute is omitted it is assumed that the attachment is a text attachment.

name="filename"
The filename. When sending a file attachment this name is what the reciever will see in his/hers list of attachment, not the original filename. If omitted, the original name will be used. This attribute is required when sending inline text or binary attachments.

mimetype="MIME type"
Sets the MIME type of the file. Since MIME type is set by the *Content types* module this setting seldom needs to be used.

mimeencoding="MIME encoding"
Sets the MIME encoding of the file. If omitted the *E-mail module's* default setting within the Roxen WebServer Administration interface might be used if it's a *file* attachment.

<header/> or <header></header>

Provided by module: *Tags: E-mail module*

This subtag/container is designed for adding additional headers to the mail.

Attributes

name="string"
The name of the header. Standard headers are 'From:', 'To:', 'Cc:', 'Bcc:' and 'Subject:'. However, there are no restrictions on how many headers are sent.

value=""
The value of the header. This attribute is only used when using the singletag version of the tag. In case of the tag being used as a containertag the content will be the value.

```
<email from="foo@bar.com" to="johny@pub.com|pity@bu
fet.com|ely@rest.com"
separator="|" charset="iso-8859-
2" server="mailhub.anywhere.org">

<header name="X-foo-header" value="one two three" /
>
<header name="Importance">Normal</header>
<header name="X-MSMail-Priority" value="Normal" />
 This is the contents.
</email>
```

<signature/>

Provided by module: *Tags: E-mail module*

This container is designed for adding a signature to the mail.

Attributes

```
<email from="foo@bar.com" to="johny@pub.com|pity@bu
fet.com|ely@rest.com"
separator="|" charset="iso-8859-
2" server="mailhub.anywhere.org">

<header name="X-foo-header" value="one two three" /
>
<header name="Importance">Normal</header>
<header name="X-MSMail-Priority" value="Normal" />
 This is the contents.

<signature>
-------------------
John Doe
Roxen Administrator
</signature>
</email>
```

<expire-time/>

Provided by module: *Tags: RXML 2 tags*

Sets client cache expire time for the document by sending the HTTP header "Expires". Note that on most systems

the time can only be set to dates before 2038 due to operating software limitations.

Attributes

now
Notify the client that the document expires now. The headers "Pragma: no-cache" and "Cache-Control: no-cache" will also be sent, besides the "Expires" header.

unix-time="number"
The exact time of expiration, expressed as a posix time integer.

years="number"
Add this number of years to the result.

months="number"
Add this number of months to the result.

weeks="number"
Add this number of weeks to the result.

days="number"
Add this number of days to the result.

hours="number"
Add this number of hours to the result.

beats="number"
Add this number of beats to the result.

minutes="number"
Add this number of minutes to the result.

seconds="number"
Add this number of seconds to the result.

<header/>

Provided by module: *Tags: RXML 2 tags*

Adds a HTTP header to the page sent back to the client. For more information about HTTP headers please steer your browser to chapter 14, 'Header field definitions' in RFC 2616, available at Roxen Community.

Attributes

name="string"
The name of the header.

value="string"
The value of the header.

<killframe/>

Provided by module: *Tags: Kill frame*

This tag adds some JavaScript that will prevent others from putting the page in a frame. It can also strip any occurrences of index files, like *index.html*, from the end of the URL.

Attributes

killindex
Removes trailing index.html from the URL.

<redirect/>

Provided by module: *Tags: RXML 2 tags*

Redirects the user to another page by sending a HTTP redirect header to the client. If the redirect is local, i.e. within the server, all prestates are preserved. E.g. "/index.html" and "index.html" preserves the prestates, while "http://server.com/index.html" does not.

Attributes

to="URL"
The location to where the client should be sent.

add="string"
The prestate or prestates that should be added, in a comma separated list.

drop="string"
The prestate or prestates that should be dropped, in a comma separated list.

drop-all
Removes all prestates from the redirect target.

text="string"
Sends a text string to the browser, that hints from where and why the page was redirected. Not all browsers will show this string. Only special clients like Telnet uses it.

Arguments prefixed with "add" or "drop" are treated as prestate toggles, which are added or removed, respectively, from the current set of prestates in the URL in the redirect header (see also <*apre*>). Note that this only works when the to=... URL is absolute, i.e. begins with a "/", otherwise these state toggles have no effect.

<remove-cookie/>

Provided by module: *Tags: RXML 2 tags*

Sets the expire-time of a cookie to a date that has already occured. This forces the browser to remove it. This tag won't remove the cookie, only set it to the empty string, or what is specified in the value attribute and change it's expire-time to a date that already has occured. This is unfortunutaly the only way as there is no command in HTTP for removing cookies. We have to give a hint to the browser and let it remove the cookie.

Attributes

name
Name of the cookie the browser should remove.

value="text"
Even though the cookie has been marked as expired some browsers will not remove the cookie until it is shut down. The text provided with this attribute will be the cookies intermediate value.

Note that removing a cookie won't take effect until the next page load.

<return/>

Provided by module: *Tags: RXML 2 tags*

Attributes

code="integer"
The HTTP status code to return.

text
The HTTP status message to set. If you don't provide one, a default message is provided for known HTTP status codes, e g "No such file or directory." for code 404.

<set-cookie/>

Provided by module: *Tags: RXML 2 tags*

Sets a cookie that will be stored by the user's browser. This is a simple and effective way of storing data that is local to the user. If no arguments specifying the time the cookie should survive is given to the tag, it will live until the end of the current browser session. Otherwise, the cookie will be persistent, and the next time the user visits the site, she will bring the cookie with her.

Note that the change of a cookie will not take effect until the next page load.

Attributes

name="string"
The name of the cookie.

seconds="number"
Add this number of seconds to the time the cookie is kept.

minutes="number"
Add this number of minutes to the time the cookie is kept.

hours="number"
Add this number of hours to the time the cookie is kept.

days="number"
Add this number of days to the time the cookie is kept.

weeks="number"
Add this number of weeks to the time the cookie is kept.

months="number"
Add this number of months to the time the cookie is kept.

years="number"
Add this number of years to the time the cookie is kept.

persistent
Keep the cookie for five years.

domain
The domain for which the cookie is valid.

value="string"
The value the cookie will be set to.

path="string" (/)
The path in which the cookie should be available. Use path="" to remove the path argument from the sent cookie, thus making the cookie valid only for the present directory and below.

<throttle/>

Provided by module: *Throttling control tags*

This tag determines a request's allocated bandwidth.

Attributes

not
Disables all and any throttling for the current request. Implies the 'final' arg.

add="rate"
Adds 'rate' bytes/sec to the current rate for the current request.

subtract="rate"
Subtracts 'rate' bytes/sec from the current rate for the current request.

multiply="float"
Multiplies this requests' bandwidth by 'float'.

divide="float"
Divides this requests' bandwidth by 'float'.

rate="value"
Sets this request's bandwidth to 'value'.

final
No subsequent modifications will be done to this request's bandwidth after the current one.

If Tags

If-tags make it possible to make dynamic pages that show different content based on conditions. Authenticated users can get confidential information and pages can be optimized for all browsers. They also makes it possible to program web applications in RXML, without using any programming language.

Learn how to use the if tags from scratch in our *If tags* tutorial.

<else></else>

Provided by module: *Tags: RXML 2 tags*

Execute the contents if the previous *<if>* tag didn't, or if there was a *<false>* tag above. This tag also detects if the page's truth value has been set to false, which occurrs whenever a runtime error is encountered. The *<emit>* tag, for one, signals this way when it did not loop a single time.

The result is undefined if there has been no *<if>*, *<true>*, *<false>* or other tag that touches the page's truth value earlier in the page.

<elseif></elseif>

Provided by module: *Tags: RXML 2 tags*

Same as the *<if>*, but it will only evaluate if the previous `<if>` returned false.

<false/>

Provided by module: *Tags: RXML 2 tags*

Internal tag used to set the return value of *If Tags*. It will ensure that the next *<else>* tag will show its contents. It can be useful if you are writing your own *<if>* lookalike tag.

<if></if>

Provided by module: *Tags: RXML 2 tags*

`<if>` is used to conditionally show its contents.The `<if>` tag is used to conditionally show its contents. *<else>* or *<elseif>* can be used to suggest alternative content.

It is possible to use glob patterns in almost all attributes, where * means match zero or more characters while ? matches one character. * Thus t*f?? will match trainfoo as well as * tfoo but not trainfork or tfo. It is not possible to use regexp's together with any of the if-plugins.

The tag itself is useless without its plugins. Its main functionality is to provide a framework for the plugins.

It is mandatory to add a plugin as one attribute. The other attributes provided are and, or and not, used for combining plugins or logical negation.

```
<if variable='var.foo > 0' and='' match='var.bar is
 No'>
    ...
  </if>
```

```
<if variable='var.foo > 0' not=''>
  &var.foo; is less than 0
</if><else>
  &var.foo; is greater than 0
</else>
```

Operators valid in attribute expressions are: '=', '==', 'is', '!=', '<' and '>'.

The If plugins are sorted according to their function into five categories: Eval, Match, State, Utils and SiteBuilder.

The Eval category is the one corresponding to the regular tests made in programming languages, and perhaps the most used. They evaluate expressions containing variables, entities, strings etc and are a sort of multi-use plugins. All If-tag operators and global patterns are allowed.

```
  <set variable='var.x' value='6'/>
<if variable='var.x > 5'>More than one hand</if>

More than one hand
```

The Match category contains plugins that match contents of something, e.g. an IP package header, with arguments given to the plugin as a string or a list of strings.

```
  Your domain <if ip='130.236.*'> is </if>
<else> isn't </else> liu.se.
  Your domain
 isn't  liu.se.
```

State plugins check which of the possible states something is in, e.g. if a flag is set or not, if something is supported or not, if something is defined or not etc.

```
     Your browser
  <if supports='javascript'>
   supports Javascript version &client.javascript;
  </if>
  <else>doesn't support Javascript</else>.
     Your browser

  doesn't support Javascript.
```

Utils are additonal plugins specialized for certain tests, e.g. date and time tests.

```
  <if time='1700' after=''>
    Are you still at work?
  </if>
  <elseif time='0900' before=''>
     Wow, you work early!
  </elseif>
```

```
<else>
 Somewhere between 9 to 5.
</else>
```

SiteBuilder plugins requires a Roxen Platform SiteBuilder installed to work. They are adding test capabilities to web pages contained in a SiteBuilder administrated site.

Attributes

not
Inverts the result (true->false, false->true).

or
If any criterion is met the result is true.

and
If all criterions are met the result is true. And is default.

<if accept></if>

Provided by module: *Tags: RXML 2 tags*

Returns true if the browser accepts certain content types as specified by it's Accept-header, for example image/jpeg or text/html. If browser states that it accepts */* that is not taken in to account as this is always untrue. Accept is a *Match* plugin.

Attributes

accept="type1[,type2,...]"

<if client></if>

Provided by module: *Tags: RXML 2 tags*

Compares the user agent string with a pattern. Client is an *Match* plugin.

Attributes

client=""

<if clientvar></if>

Provided by module: *Tags: RXML 2 tags*

Evaluates expressions with client specific values. Clientvar is an *Eval* plugin.

Attributes

clientvar="variable [is value]"
Choose which variable to evaluate against. Valid operators are '=', '==', 'is', '!=', '<' and '>'.

Available variables are:

<if config></if>

Provided by module: *Tags: RXML 2 tags*

Has the config been set by use of the tag? Config is a *State* plugin.

Attributes

config="name"

<if cookie></if>

Provided by module: *Tags: RXML 2 tags*

Does the cookie exist and if a value is given, does it contain that value? Cookie is an *Eval* plugin.

Attributes

cookie="name[is value]"

<if date></if>

Provided by module: *Tags: RXML 2 tags*

Is the date yyyymmdd? The attributes before, after and inclusive modifies the behavior. Date is a *Utils* plugin.

Attributes

date="yyyymmdd"
Choose what date to test.

after
The date after todays date.

before
The date before todays date.

inclusive
Adds todays date to after and before.

```
 <if date='19991231' before='' inclusive=''>
  - 19991231
</if>
<else>
  20000101 -
</else>
```

```
 20000101 -
```

<if defined></if>

Provided by module: *Tags: RXML 2 tags*

Tests if a certain RXML define is defined by use of the <*define*> tag. Defined is a *State* plugin.

Attributes

defined="define"
Choose what define to test.

<if domain></if>

Provided by module: *Tags: RXML 2 tags*

Does the user's computer's DNS name match any of the patterns? Note that domain names are resolved asynchro-

nously, and that the first time someone accesses a page, the domain name will probably not have been resolved. Domain is a *Match* plugin.

Attributes

domain="pattern1[,pattern2,...]"
Choose what pattern to test.

<if exists></if>

Provided by module: *Tags: RXML 2 tags*

Returns true if the named page is viewable. A non viewable page is e.g. a file that matches the internal files patterns in the filesystem module. If the path does not begin with /, it is assumed to be a URL relative to the directory containing the page with the `<if>`-statement. 'Magic' files like /internal-roxen-unit will evaluate as true.

Attributes

exists="path"
Choose what path in the virtual filesystem to test.

<if expr></if>

Provided by module: *Tags: RXML 2 tags*

This plugin evaluates a string as a pike expression. Available arithmetic operators are +, -, *, / and % (modulo). Available relational operators are <, >, ==, !=, <= and >=. Available bitwise operators are &, | and ^, representing AND, OR and XOR. Available boolean operators are && and ||, working as the pike AND and OR.

Numbers can be represented as decimal integers when numbers are written out as normal, e.g. "42". Numbers can also be written as hexadecimal numbers when precedeed with "0x", as octal numbers when precedeed with "0" and as binary number when precedeed with "0b". Numbers can also be represented as floating point numbers, e.g. "1.45" or "1.6E5". Numbers can be converted between floats and integers by using the cast operators "(float)" and "(int)".

```
(int)3.14
```

A common problem when dealing with variables from forms is that a variable might be a number or be empty. To ensure that a value is produced the special functions INT and FLOAT may be used. In the expression "INT()+1" the INT-function will produce 0 if the form variable is empty, hence preventing the incorrect expression "+1" to be produced.

Attributes

expr="expression"
Choose what expression to test.

<if false></if>

Provided by module: *Tags: RXML 2 tags*

This will always be true if the truth value is set to be false. Equivalent with *<else>*. False is a *State* plugin.

Attributes

false
Show contents if truth value is true.

<if group></if>

Provided by module: *Tags: RXML 2 tags*

Checks if the current user is a member of the group according the groupfile. Group is a *Utils* plugin.

Attributes

group="name"
Choose what group to test.

groupfile="path"
Specify where the groupfile is located.

<if internal-exists></if>

Provided by module: *Tags: RXML 2 tags*

Returns true if the named page exists. If the page at the given path is non-viewable, e.g. matches the internal files patterns in the filesystem module, it will still be detected by this if plugin. If the path does not begin with /, it is assumed to be a URL relative to the directory containing the page with the if statement. 'Magic' files like /internal-roxen-unit will evaluate as true.

Attributes

internal-exists="path"
Choose what path in the virtual filesystem to test.

<if ip></if>

Provided by module: *Tags: RXML 2 tags*

Does the users computers IP address match any of the patterns? This plugin replaces the Host plugin of earlier RXML versions. Ip is a *Match* plugin.

Attributes

ip="pattern1[,pattern2,...]"
Choose what IP-adress pattern to test.

<if language></if>

Provided by module: *Tags: RXML 2 tags*

Does the client prefer one of the languages listed, as specified by the Accept-Language header? Language is a *Match* plugin.

Attributes

language="language1[,language2,...]"
Choose what language to test.

<if match></if>

Provided by module: *Tags: RXML 2 tags*

Evaluates patterns. More information can be found in the *If tags tutorial*. Match is an *Eval* plugin.

Attributes

match="pattern"
Choose what pattern to test. The pattern could be any expression. Note!: The pattern content is treated as strings:

```
   <set variable='var.hepp' value='10' />

 <if match='var.hepp is 10'>
  true
 </if>
 <else>
  false
 </else>



  false
```

This example shows how the plugin treats "var.hepp" and "10" as strings. Hence when evaluating a variable as part of the pattern, the entity associated with the variable should be used, i.e. var.hepp instead of var.hepp. A correct example would be:

```
  <set variable='var.hepp' value='10' />

 <if match='&var.hepp; is 10'>
  true
 </if>
 <else>
  false
 </else>



  true
```

Here, is treated as an entity and parsed correctly, letting the plugin test the contents of the entity.

<if Match></if>

Provided by module: *Tags: RXML 2 tags*

Case sensitive version of the match plugin

<if module></if>

Provided by module: *Tags: RXML 2 tags*

Returns true if the selected module is enabled in the current server. This is useful when you are developing RXML applications that you plan to move to other servers, to ensure that all required modules are added.

Attributes

module="name"
The "real" name of the module to look for, i.e. its filename without extension and without directory path.

<if pragma></if>

Provided by module: *Tags: RXML 2 tags*

Compares the HTTP header pragma with a string. Pragma is a *State* plugin.

Attributes

pragma="string"
Choose what pragma to test.

```
   <if pragma='no-
cache'>The page has been reloaded!</if>
 <else>Reload this page!</else>

 Reload this page!
```

<if prestate></if>

Provided by module: *Tags: RXML 2 tags*

Are all of the specified prestate options present in the URL? Prestate is a *State* plugin.

Attributes

prestate="option1[,option2,...]"
Choose what prestate to test.

<if referrer></if>

Provided by module: *Tags: RXML 2 tags*

Does the referrer header match any of the patterns? Referrer is a *Match* plugin.

Attributes

referrer="pattern1[,pattern2,...]"
Choose what pattern to test.

<if sizeof></if>

Provided by module: *Tags: RXML 2 tags*

Compares the size of a variable with a number.

```
  <set variable="var.x" value="hello"/>
<set variable="var.y" value=""/>
<if sizeof="var.x == 5">Five</if>
<if sizeof="var.y > 0">Nonempty</if>


Five
```

<if supports></if>

Provided by module: *Tags: RXML 2 tags*

Does the browser support this feature? Supports is a *State* plugin.

Attributes

supports="feature"
Choose what supports feature to test.

The following features are supported:

<if time></if>

Provided by module: *Tags: RXML 2 tags*

Is the time hhmm? The attributes before, after and inclusive modifies the behavior. Time is a *Utils* plugin.

Attributes

time="hhmm"
Choose what time to test.

after
The time after present time.

before
The time before present time.

inclusive
Adds present time to after and before.

```
<if time='1200' before='' inclusive=''>
  ante meridiem
</if>
<else>
  post meridiem
</else>
```

<if true></if>

Provided by module: *Tags: RXML 2 tags*

This will always be true if the truth value is set to be true. Equivalent with <*then*>. True is a *State* plugin.

Attributes

true
Show contents if truth value is false.

<if user></if>

Provided by module: *Tags: RXML 2 tags*

Has the user been authenticated as one of these users? If any is given as argument, any authenticated user will do. User is a *Utils* plugin.

Attributes

user="{name1[,name2,...], any}"
Specify which users to test.

<if variable></if>

Provided by module: *Tags: RXML 2 tags*

Does the variable exist and, optionally, does it's content match the pattern? Variable is an *Eval* plugin.

Attributes

variable="name[operator pattern]"
Choose variable to test. Valid operators are '=', '==', 'is', '!=', '<' and '>'.

<if Variable></if>

Provided by module: *Tags: RXML 2 tags*

Case sensitive version of the variable plugin

<then></then>

Provided by module: *Tags: RXML 2 tags*

Shows its content if the truth value is true. This is useful in conjunction with tags that leave status data there, such as the <*emit*> or <*crypt*> tags.

<true/>

Provided by module: *Tags: RXML 2 tags*

An internal tag used to set the return value of *If Tags*. It will ensure that the next <*else*> tag will not show its contents. It can be useful if you are writing your own <*if*> lookalike tag.

Flow Tags

Flow tags control and guide the flow of RXML code through conditions and error checking. Like the *If Tags* these tags can be used to create advanced dynamic pages. These tags are unlike the If Tags as they do not take any attributes that process or find information, only attributes that affect the tag itself.

The tags <*throw*> and <*catch*> for instance, are used to help programmers more easily debug their code.

<catch></catch>

Provided by module: *Tags: RXML 2 tags*

Evaluates the RXML code, and, if nothing goes wrong, returns the parsed contents. If something does go wrong, the error message is returned instead. See also <*throw*>.

<cond></cond>

Provided by module: *Tags: RXML 2 tags*

This tag makes a boolean test on a specified list of cases. This tag is almost eqvivalent to the <*if*>/<*else*> combination. The main difference is that the `<default>` tag may be put whereever you want it within the `<cond>` tag. This will of course affect the order the content is parsed. The `<case>` tag is required.

<case></case>

Provided by module: *Tags: RXML 2 tags*

This tag takes the argument that is to be tested and if it's true, it's content is executed before exiting the `<cond>`. If the argument is false the content is skipped and the next `<case>` tag is parsed.

Attributes

```
<cond>
 <case variable='form.action = edit'>
  some database edit code
 </case>
 <case variable='form.action = delete'>
  some database delete code
 </case>
 <default>
  view something from the database
 </default>
</cond>
```

<default></default>

Provided by module: *Tags: RXML 2 tags*

The `<default>` tag is eqvivalent to the `<else>` tag in an `<if>` statement. The difference between the two is that the `<default>` may be put anywhere in the `<cond>` statement. This affects the parseorder of the statement. If the `<default>` tag is put first in the statement it will allways be executed, then the next `<case>` tag will be executed and perhaps add to the result the `<default>` performed.

<for></for>

Provided by module: *Tags: RXML 2 tags*

Makes it possible to create loops in RXML.

Attributes

from="number"
Initial value of the loop variable.

step="number"
How much to increment the variable per loop iteration. By default one.

to="number"
How much the loop variable should be incremented to.

variable="name"
Name of the loop variable.

<throw></throw>

Provided by module: *Tags: RXML 2 tags*

Throws a text to be caught by <*catch*>. Throws an exception, with the enclosed text as the error message. This tag has a close relation to <*catch*>. The RXML parsing will stop at the `<throw>` tag.

Graphics Tags

Roxen WebServer features an advanced graphics engine. Together with output from a database and/or data from the live server the graphics engine can present information that makes a site feel alive. The graphics engine can be set up to automatically generate very nice-looking headers, diagrams, buttons, thumbnails and many other web graphics :

Images
Through <*cimg*> images can be transformed and converted to any of the almost 20 image-formats available.

Business Graphics
Generate online reports dynamically with <*diagram*>, represented as line, bar, pie charts and many other diagrams.

Graphical Headers
Roxen WebServer supports many font-formats and makes it easy to import them. By combining fonts with images and graphical effects very eyecatching headers can be made. For instance, it is possible to set the ordinary HTML-tag `<h1>` to output a graphical header instead of ordinary boring headers. Graphical headers can be made with <*gtext*> and <*gh*>.

Good looking graphics are an important part of the layout of web pages. Creating graphics can also be very demanding of the site's hardware, especially if it involves transforming, converting and rendering of a lot of images or graphics. It is also important to not overdo the website's layout else people with slow connections will not be able to enjoy it since it will take forever to download it.

File Formats

Some of the tags take images as attributes. For example, to use as a background. The images can be of most formats including GIF, JPEG, PSD and PNG.

Color attributes

Color attributes in RXML tags can be specified in one of the following ways. Note that ordinary HTML and XHTML tags only supports hexadecimal RGB values and a very limited array of textual names. By using the HTML color wiretap module with the setting "Normalize colors in parsed tags" set to yes it is possible to use RXML way of describing colors, at the cost of some performance.

Name
For example *black* or *darkred*.

#RGB value
The color is specified as a hexadecimal-digits, `#RRGGBB`. For example, *#ffdead* or *#00ff00*.

@HSV value
The color is specified with the syntax `@h,s,v` where h is the hue specified as degrees (0 to 359), s is the saturation specified as a percentage and v the value also specified as a percentage. For example, *@150,70,70*.

%CMYK value
The color is specified with the syntax `%c,m,y,k` where all the values are percentages. For example, *%10,20,30,40*.

Color table

Below follows a list of all defined color names and its associated RGB, HSV and CMYK vale.

Internal Images

At the location `/internal-roxen-*` in your webserver you'll find some images which are always available to be included in image tags and the like. The most useful of these are /internal-roxen-unit, which is a 1x1 pixels big transparent GIF image. Small transparent images are often used by HTML layout designers to create spaces between other elements.

internal-roxen-pixel

At the location `/internal-roxen-pixel-*` you'll find a 1x1 nontransparent pixel in the selected color. Examples are `/internal-roxen-pixel-blue` or `/internal-roxen-pixel-ff34c0`. The colors are decoded as describe in *Color attributes*.

Bitmap Files

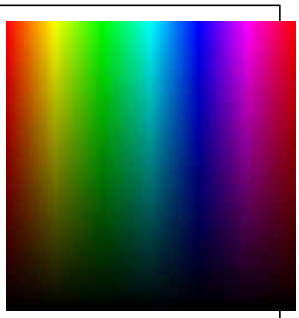
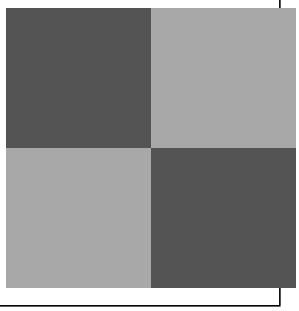
All images that reside in the directory server/roxen-images in your Roxen WebServer distribution is available through `/internal-roxen-*`. All the images available in the `server/roxen-images/dir` directory is available through `/internal-gopher-*`. E.g. if you want to put the image help.gif on your page, just use `<img src="/internal-roxen-help" />`. Note that the look of the images as well as the contents of server/roxen-images may vary.

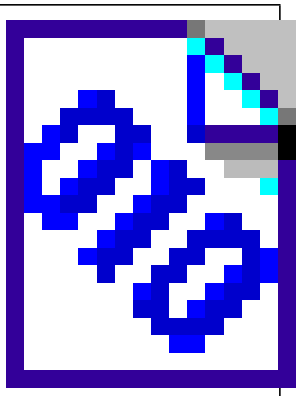
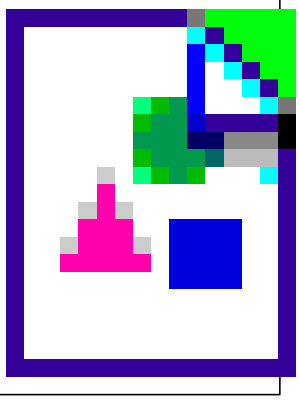
The following images will always exist with the same look and size:

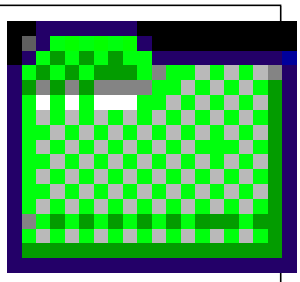
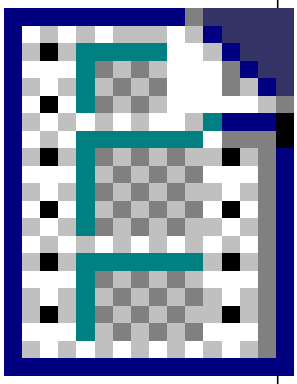
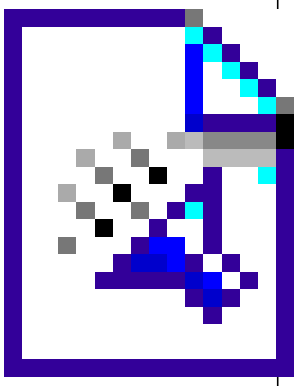
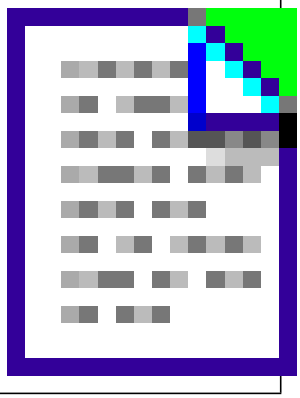
The following images will always exist, but may change in appearence.

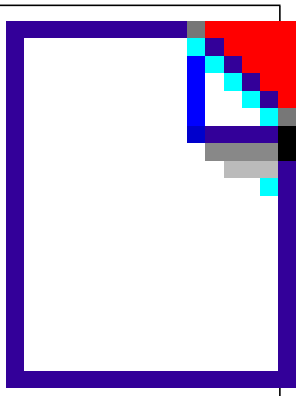
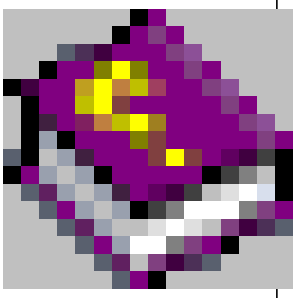
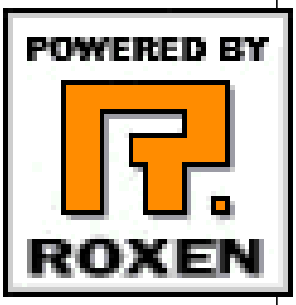
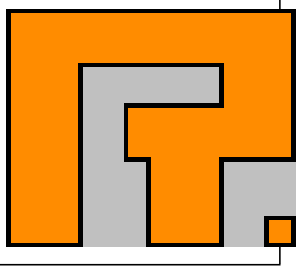
Color selector

In order to easily create a multistage color selection widget you can acces automatically generated saturation bars through the `/internal-roxen-colorbar` URL. `/internal-`

/internal-roxen-colsel	
/internal-roxen-colsel-small	
/internal-roxen-squares	

/internal-gopher-binary	
/internal-gopher-image	

/internal-gopher-menu	
/internal-gopher-movie	
/internal-gopher-sound	
/internal-gopher-text	

/internal-gopher-unknown	
/internal-roxen-help	
/internal-roxen-pike	
/internal-roxen-power	
/internal-roxen-roxen	

`roxen-colorbar:h,b,w` returns a 30x256 pixels big color bar with the hue h,the brightness b and a white marker at the location w, 0 being the bottom and 255 the top. The interval for all three variables are 0-255. The intended use is to first present the user with a two-dimensional hue/brightness selector, as shown above. The selected combination is then used as basis for which colorbar to show, where the user can selected the preferred saturation.

```
<for variable="var.i" from="0" to="255" step="24">
<img border="1" src="/internal-roxen-
colorbar:&var.i;,255,&var.i;" />
</for>
```

```
<img border="1" src="bar1.jpg" />
<img border="1" src="bar2.jpg" />
<img border="1" src="bar3.jpg" />
<img border="1" src="bar4.jpg" />
<img border="1" src="bar5.jpg" />
<img border="1" src="bar6.jpg" />
<img border="1" src="bar7.jpg" />
<img border="1" src="bar8.jpg" />
<img border="1" src="bar9.jpg" />
<img border="1" src="bar10.jpg" />
<img border="1" src="bar11.jpg" />
```

The code above will generate the 11 colorbars shown below. All of them has brighness set to 255, but their hue will range from 0 to 255 in with steps of 24.

<anfang></anfang>

Provided by module: *Graphics: Graphic text*

Creates an anfang in the beginning of a text. This tag takes the same attributes as *<gtext>*.

```
  <anfang crop="">This is a beginning<br />
 of a very short text,<br />
 and here it ends.
 </anfang>
  <anfang crop="">This is a beginning<br />
 of a very short text,<br />
 and here it ends.
 </anfang>
```

Attributes

verbatim
By default the gtext tag will try to make typographical enhancements to the text to make the resulting image as eye pleasing as possible. If you want to turn this feature off, add this attribute to the tag.

alpha="path"
Use the specified image as an alpha channel, together with the background attribute.

background="path"
Specifies the image to use as background.

tile
Tiles the background and foreground images if they are smaller than the actual image.

mirrortile
Tiles the background and foreground images around x-axis and y-axis for odd frames, creating seamless textures.

bevel="width"
Draws a bevel-box around the text.

bgcolor="color"
Sets the background color. If the module "HTML color wiretap" is loaded, they are taken from the normal HTML tags in your document, like body, table, tr and td.
If you set the background color, it is probably best to add the notrans attribute as well.

bgturbulence="frequency,color;frequency,color..."

Apply a turbulence effect on the background.

bold
Use a bold version of the font, if available. Can not be used together with the black or light attributes.

black
Use a black, or heavy, version of the font, if available. Can not be used together with the bold or light attributes.

light
Use a light version of the font, if available. Can not be used together with the bold or black attributes.

italic
Use an italic version of the font, if available.

bshadow="distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible, but you might have to add 'spacing'.

chisel
Make the text look like it has been cut into the background.

crop
Remove all white-space around the image.

encoding="string"
Choose with which charset the text is encoded with.

fadein="blur,steps,delay,initialdelay"
Generates an animated GIF file of a fade-in effect.

fgcolor="color"
Sets the text color.

font="string"
Selects which font to use. You can get a list of all available fonts by using the list fonts task in the administration interface, or by using the <*emit fonts*> plugin.

fontsize="number"
Selects which size of the font that should be used.

format="string"
Set the image format, e.g. "png".

fs
Apply floyd-steinberg dithering to the resulting image. Most of the time it is much better to increase the number of colors, instead of dithering the image, but sometimes when using very complex background images dithering is O.K.

ghost="dist,blur,color"
Apply a ghost effect. Cannot be used together with shadow or magic coloring.

glow="color"
Apply a 'glow' filter to the image. Quite a CPU eater. Looks much better on a dark background, where a real 'glow' effect can be achieved.

maxlen="number"
Sets the maximum length of the text that will be rendered into an image, by default 300.

move="x,y"
Moves the text relative to the upper left corner of the background image. This will not change the size of the image.

narrow
Use a narroe version of the font, if available.

notrans
Do not make the background transparent. Useful when making 'boxes' of color around the text.

nowhitespace
Removes all whitespaces before and after the real text.

opaque="percentage"
Sets the 'opaque' value of the color used to draw the text. Default is 100%. In the example below, notice how the text color mixes with the two background colors.

outline="color,extra-radius"
Draw an outline around the text. Quite useful when combined with textscale.

pressed
Inverts the direction of the bevel box, to make it look like a button that is pressed down. The magic modifier will do this automatically.

quant="number"
Quantifies the image with this number of colors. Using a lower number will decrease the image (file)size, but make the text look more 'edgy', and if you use complex backgrounds or image textures, more colors will be neded. At most 255 colors can be used, and less than 2 is quite useless. It is advisable to use powers of 2 to optimize the palette allocation.

rescale
Rescale the background to fill the whole image.

rotate="angle"
Rotates the image this number of degrees counter-clockwise.

scale="number"
Sets the scale of the image. Larger than 1.0 is enlargement.

scolor="color"
Use this color for the shadow. Used with the shadow attribute.

scroll="width,steps,delay"
Generate an animated GIF image of the text scrolling.

shadow="intensity,distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible,

size="width,height"
Set the size of the image.

spacing="number"
Add space around the text.

talign="{left, right, center}"
Adjust the alignment of the text.

textbelow="color"
Place the text centered in a box of the given color below the image area. Useful together with background to make captions for images.

textbox="opaque,color"

Draw a box with an opaque value below the text of the specified color.

textscale="color,color,color,color"
Apply a color filter to the text. The colors are, respectively, upper left, lower left, upper right and lower right. It is probably a good idea to increase the 'quant' value when using this argument.

texture="path"
Uses the specified images as a field texture.

tile
Tiles the background and foreground images if they are smaller than the actual image.

xpad="{percentage, integer ended with "px"}"
Sets the padding between characters. The value can either be an relative change, in percent, or an absolute number of pixels, ended with the string "px". Note that different fonts reacts differently on these values and for some it will not have any effect at all. This depends on the type of the font and the font implementation.

xsize="number"
Sets the width.

xspacing="number"
Sets the horizontal spacing.

ypad="{percentage, integer ended with "px"}"
Sets the padding beteen lines. Works as xpad.

ysize="number"
Sets the height.

yspacing="number"
Sets the vertical spacing.

<atlas></atlas>

Provided by module: *Graphics: Atlas*

Draws a map. The map shows either the world, regions (Africa, Europe, etc) or countries. It's a known bug that the map is not entierly up to date.

```
  <atlas/>
  <atlas/>
```

```
  <atlas fgcolor='#425A84' bgcolor='#dee2eb'>
<country domain='se' color='orange'/>
<country domain='jp' color='orange'/>
<marker x='100' y='90'/>
</atlas>
  <atlas fgcolor='#425A84' bgcolor='#dee2eb'>
<country domain='se' color='orange'/>
<country domain='jp' color='orange'/>
<marker x='100' y='90'/>
</atlas>
```

Attributes

region="name" (The World)
Which map to show. The value may be any of the listed region values that emit plugin *atlas* returns.

```
  <atlas region='europe' width='200'/>
```

```
  <atlas region='europe' width='200'/>
```

width="number"
The width of the image.

height="number"
The height of the image.

fgcolor="color" (white)
The color of the unselected land areas.

bgcolor="color" (#101040)
The color of the sea areas.

<country/>

Provided by module: *Graphics: Atlas*

A region that should be highlighted with a different color on the map.

Attributes

domain="name"
The top domain of the country that should be highlighted.

name="name"
The name of the country that should be highlighted. A list of available names can be aquired from the *atlas* emit plugin.

color="color" (#e0c080)
The color that should be used for highlighting.

<marker/>

Provided by module: *Graphics: Atlas*

Draws a marker at the specified position

Attributes

x="pixels or percentage"
The distance from the left of the map.

y="pixels or percentage"
The distance from the top of the map.

color="color" (red)
The color of the marker

style="{box, diamond}" (diamond)
The type of marker.

```
  <atlas region='europe' width='150'>
<marker x='100' y='30' style='diamond' />
<marker x='125' y='30' style='box' />
</atlas>
  <atlas region='europe' width='150'>
<marker x='100' y='30' style='diamond' />
<marker x='125' y='30' style='box' />
</atlas>
```

size="number" (4)
The size of the marker.

<cimg/>

Provided by module: *Graphics: Image converter*

Manipulates and converts images between different image formats. Provides the tag `<cimg>` that makes it is possible to convert, resize, crop and in other ways transform images.

Attributes

src="url"
The path to the indata file.

```
<cimg src='/internal-roxen-testimage'/>
<cimg src='/internal-roxen-testimage'/>
```

data="imagedata"
Insert images from other sources, e.g. databases through entities or variables.

```
<emit source='sql' query='select imagedata from ima
ges where id=37'>
<cimg data='&sql.imagedata;'/>
</emit>
```

format="{gif, jpeg, png, avs, gmp, bd, hrz, ilbm, psx, pnm, ps, pvr, tga, tiff, wbf, xbm, xpm}" (gif)
The format to encode the image to. The formats available are:

Acronym	Acronym interpretation
gif	Graphics Interchange Format (might be missing in your roxen)
jpeg	Joint Photography Expert Group image compression
png	Portable Networks Graphics
avs	Advanced Visual Systems Inc. image format
bmp	Windows BitMaP file
gd	Internal format used by libgd
hrz	HRZ is (was?) used for amatuer radio slow-scan TV.
ilbm	Interchangeable File Format: interleaved bitmap
pcx	Zsoft PCX file format (PC / DOS)
pnm	Portable AnyMap
ps	Adobe PostScript file
pvr	Pover VR (dreamcast image)
tga	TrueVision Targa (PC / DOS)
tiff	Tag Image File Format
wbf	WAP Bitmap File
xbm	XWindows Bitmap File
xpm	XWindows Pixmap File

```
<cimg src='/internal-roxen-
testimage' format='png'/>
<cimg src='/internal-roxen-
testimage' format='png'/>
```

```
<cimg src='/internal-roxen-
testimage' format='gif'/>
<cimg src='/internal-roxen-
testimage' format='gif'/>
```

quant="number" (format dependant)
The number of colors to quantizize the image to.
Default for gif is 32(+1 transparent), for most other formats (except black and white) is it unlimited.

```
<cimg src='/internal-roxen-testimage' quant='2'/>
<cimg src='/internal-roxen-testimage' quant='2'/>
```

dither="{none, random, floyd-steinberg}" (none)
Choose the dithering method.

Method	Meaning
none	No dithering is performed at all.
random	Random scatter dither. Not visually pleasing, but it is useful for very high resolution printing.
floyd-steinberg	Error diffusion dithering. Usually the best dithering method.

```
<cimg src='/internal-roxen-
testimage' dither='random' quant='10'/>
<cimg src='/internal-roxen-
testimage' dither='random' quant='10'/>
```

```
<cimg src='/internal-roxen-
testimage' dither='floyd-steinberg' quant='10'/>
<cimg src='/internal-roxen-
testimage' dither='floyd-steinberg' quant='10'/>
```

true-alpha
If present, render a real alpha channel instead of on/off alpha. If the file format only supports on/off alpha, the alpha channel is dithered using a floyd-steinberg dither.

```
<cimg src='/internal-roxen-testimage' opaque-
value='20'/>
<cimg src='/internal-roxen-testimage' opaque-
```

```
value='20'/>
```

```
  <cimg src='/internal-roxen-testimage' opaque-
value='20' true-alpha='1'/>
  <cimg src='/internal-roxen-testimage' opaque-
value='20' true-alpha='1'/>
```

background-color="color" (taken from the page)
The color to render the image against.

```
  <cimg src='/internal-roxen-testimage' background-
color='red' opaque-value='50'/>
  <cimg src='/internal-roxen-testimage' background-
color='red' opaque-value='50'/>
```

opaque-value="percentage" (100)
The transparency value to use, 100 is fully opaque, and 0 is fully transparent.

cs-rgb-hsv="{0, 1}" (0)
Perform rgb to hsv colorspace conversion.

```
  <cimg src='/internal-roxen-testimage' cs-rgb-
hsv='1'/>
  <cimg src='/internal-roxen-testimage' cs-rgb-
hsv='1'/>
```

gamma="number" (1.0)
Perform gamma adjustment.

```
  <cimg src='/internal-roxen-
testimage' gamma='0.5'/>
  <cimg src='/internal-roxen-
testimage' gamma='0.5'/>
```

```
  <cimg src='/internal-roxen-
testimage' gamma='1.5'/>
  <cimg src='/internal-roxen-
testimage' gamma='1.5'/>
```

cs-grey="{0, 1}" (0)
Perform rgb to greyscale colorspace conversion.

```
  <cimg src='/internal-roxen-testimage' cs-
grey='1'/>
  <cimg src='/internal-roxen-testimage' cs-
grey='1'/>
```

cs-invert="{0, 1}" (0)
Invert all colors

```
  <cimg src='/internal-roxen-testimage' cs-
invert='1'/>
  <cimg src='/internal-roxen-testimage' cs-
invert='1'/>
```

cs-hsv-rgb="{0, 1}" (0)
Perform hsv to rgb colorspace conversion.

```
  <cimg src='/internal-roxen-testimage' cs-hsv-
rgb='1'/>
  <cimg src='/internal-roxen-testimage' cs-hsv-
rgb='1'/>
```

rotate-cw="degree" (0)
Rotate the image clock-wise.

```
  <cimg src='/internal-roxen-testimage' rotate-
cw='20'/>
  <cimg src='/internal-roxen-testimage' rotate-
cw='20'/>
```

rotate-ccw="degree" (0)
Rotate the image counter clock-wise.

rotate-unit="{rad, deg, ndeg, part}" (deg)
Select the unit to use while rotating.

Unit	Meaning
rad	Radians
deg	Degrees
ndeg	'New' degrees (400 for each full rotation)
part	0 - 1.0 (1.0 == full rotation)

mirror-x="{0, 1}" (0)
Mirror the image around the X-axis.

mirror-y="{0, 1}" (0)
Mirror the image around the Y-axis.

scale="fact" (1.0)
Scale fact times. (0.5 -> half size, 2.0 -> double size)

```
  <cimg src='/internal-roxen-
testimage' scale='0.5'/>
  <cimg src='/internal-roxen-
testimage' scale='0.5'/>
```

scale="x,y"
Scale to the exact size x,y. If either of X or Y is zero, the image is scaled to the specified width or hight, and the value that is zero is scaled in proportion to the other value.

```
  <cimg src='/internal-roxen-
testimage' scale='20,50'/>
  <cimg src='/internal-roxen-
testimage' scale='20,50'/>
```

max-width="xsize"
If width is larger than 'xsize', scale width to 'xsize' while keeping aspect.

max-height="ysize"
If height is larger than 'ysize', scale height to 'ysize' while keeping aspect.

span-width="xsize"
If width is larger than 'xsize', scale width to 'xsize' while keeping aspect. If width is smaller than 'xsize', extend width to 'xsize' by filling the new space with current background color.

```
  <cimg src='/internal-roxen-testimage' span-
width="350" background-color='white'/>
  <cimg src='/internal-roxen-testimage' span-
```

```
width="350" background-color='white'/>
```

span-height="ysize"
If height is larger than 'ysize', scale height to 'ysize' while keeping aspect. If height is smaller than 'ysize', extend height to 'ysize' by filling the new space with current background color.

```
  <cimg src='/internal-roxen-testimage' span-
height="350" background-color='white'/>
  <cimg src='/internal-roxen-testimage' span-
height="350" background-color='white'/>
```

x-offset="pixels" (0)
Cut n pixels from the beginning of the X scale.

```
  <cimg src='/internal-roxen-testimage' x-
offset='100'/>
  <cimg src='/internal-roxen-testimage' x-
offset='100'/>
```

y-offset="pixels" (0)
Cut n pixels from the beginning of the Y scale.

x-size="pixels" (whole image)
Keep n pixels from the beginning of the X scale.

```
  <cimg src='/internal-roxen-testimage' x-
size='100'/>
  <cimg src='/internal-roxen-testimage' x-
size='100'/>
```

y-size="pixels" (whole image)
Keep n pixels from the beginning of the Y scale.

crop="{x1,y1-x2,y2, auto, guides-cross, guides-region}"
Crops the image by using several differen methods. The simplest is to only specify the area to be cropped with x,y-x,y coordinates. By instead selecting "auto" the image will be cropped so that as many pixels as possible with the same color is removed from around the image.

A more advanced cropping method can be used by giving the crop argument "guides-cross". The image will then be cropped around the intersection of two guides inside the image. Guides can be added to e.g. Photoshop and GIMP images. If several guides are present, which ones to use can be selected with the guides-index=x,y attribute, where x and y is the number of the guides. Guides cross cropping is usefull together with max-width and max-height attributes when creating thumb nails.

A combination of guides cross cropping can be used by giving the crop argument "guides-region". In this cropping mode the area enclosed by two horizontal and two vertical guides are saved. Which guides to use is given by the guides-index=x1,y1-x2,y2 attribute, where the x and y parameters are the number of the guides. Guides can also be specified as guides-index=x,y. Then the saved area will be the one enclosed by the buides x,y and x+1,y+1, counting from left and top. Again, combine max-width and max-height makes a good effect, since scaling is performed after cropping.

```
  <cimg src='/internal-roxen-
testimage' crop='50,28-150,92'/>
  <cimg src='/internal-roxen-
testimage' crop='50,28-150,92'/>
```

jpeg-quality="percentage" (75)
Set the quality on the output jpeg image.

```
  <cimg src='/internal-roxen-
testimage' format='jpeg' jpeg-quality='30'/>
  <cimg src='/internal-roxen-
testimage' format='jpeg' jpeg-quality='30'/>
```

```
  <cimg src='/internal-roxen-
testimage' format='jpeg' jpeg-quality='1'/>
  <cimg src='/internal-roxen-
testimage' format='jpeg' jpeg-quality='1'/>
```

jpeg-optimize="{0, 1}" (1)
If 0, do not generate optimal tables. Somewhat faster, but produces bigger files.

jpeg-progressive=="{0, 1}" (0)
Generate progressive jpeg images.

jpeg-smooth="0-100" (0)
Smooth the image while compressing it. This produces smaller files, but might undo the effects of dithering.

bmp-bpp="1,4,8,24" (24)
Force this number of bits per pixel for bmp images.

bmp-windows="{0, 1}" (1)
Windows or OS/2 mode, default is 1. (windows mode)

bmp-rle="{0, 1}" (0)
RLE 'compress' the BMP image.

gd-alpha_index="color" (0)
Color in the colormap to make transparent for GD-images with alpha channel.

pcx-raw="{1, 0}" (0)
If 1, do not RLE encode the PCX image.

pcx-dpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-xdpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-ydpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-xoffset="0-imagexsize-2" (0)
Offset from start of image data to image content for PCX images. Unused by most programs.

pcx-yoffset="0-imageysize-2" (0)
Offset from start of image data to image content for PCX images. Unused by most programs.

tga-raw="{1, 0}" (0)
If 1, do not RLE encode the Targa image.

ps-dpi="0-10000000.0" (75.0)
Dots per inch for the resulting postscript file.

<cimg-url/>

Provided by module: *Graphics: Image converter*

This tag generates an URL to the manipulated picture. `<cimg-url>` takes the same attributes as *<cimg>*, including the image cache attributes. The use for the tag is to insert image-URLs into various places, e.g. a submit-box.

Attributes

src="url"

The path to the indata file.

```
<cimg-url src='/internal-roxen-testimage'/>
/_internal/cimg!0/amacxg4qbc0
```

data="imagedata"

Insert images from other sources, e.g. databases through entities or variables.

```
<emit source='sql' query='select imagedata from ima
ges where id=37'>
  <cimg-url data='&sql.imagedata;'/>
</emit>
```

format="{gif, jpeg, png, avs, gmp, bd, hrz, ilbm, psx, pnm, ps, pvr, tga, tiff, wbf, xbm, xpm}" (gif)

The format to encode the image to. The formats available are:

Acronym	Acronym interpretation
gif	Graphics Interchange Format (might be missing in your roxen)
jpeg	Joint Photography Expert Group image compression
png	Portable Networks Graphics
avs	Advanced Visual Systems Inc. image format
bmp	Windows BitMaP file
gd	Internal format used by libgd
hrz	HRZ is (was?) used for amatuer radio slow-scan TV.
ilbm	Interchangeable File Format: interleaved bitmap
pcx	Zsoft PCX file format (PC / DOS)
pnm	Portable AnyMap
ps	Adobe PostScript file
pvr	Pover VR (dreamcast image)
tga	TrueVision Targa (PC / DOS)
tiff	Tag Image File Format
wbf	WAP Bitmap File
xbm	XWindows Bitmap File
xpm	XWindows Pixmap File

```
<cimg-url src='/internal-roxen-
testimage' format='png'/>
/_internal/cimg!0/amacxg4qbc0
```

```
<cimg-url src='/internal-roxen-
testimage' format='gif'/>
/_internal/cimg!0/amacxg4qbc3
```

quant="number" (format dependant)

The number of colors to quantizize the image to.

Default for gif is 32(+1 transparent), for most other formats (except black and white) is it unlimited.

```
<cimg-url src='/internal-roxen-
testimage' quant='2'/>
/_internal/cimg!0/amacxg4qbc2
```

dither="{none, random, floyd-steinberg}" (none)

Choose the dithering method.

Method	Meaning
none	No dithering is performed at all.
random	Random scatter dither. Not visually pleasing, but it is useful for very high resolution printing.
floyd-steinberg	Error diffusion dithering. Usually the best dithering method.

```
<cimg-url src='/internal-roxen-
testimage' dither='random' quant='10'/>
/_internal/cimg!0/amacxg4qbcd
```

```
<cimg-url src='/internal-roxen-
testimage' dither='floyd-steinberg' quant='10'/>
/_internal/cimg!0/amacxg4qbcc
```

true-alpha

If present, render a real alpha channel instead of on/off alpha. If the file format only supports on/off alpha, the alpha channel is dithered using a floyd-steinberg dither.

```
<cimg-url src='/internal-roxen-testimage' opaque-
value='20'/>
/_internal/cimg!0/am96m89rxt0
```

```
<cimg-url src='/internal-roxen-testimage' opaque-
value='20' true-alpha='1'/>
/_internal/cimg!0/am96ia7jtuu
```

background-color="color" (taken from the page)
The color to render the image against.

```
<cimg-url src='/internal-roxen-testimage' background-color='red' opaque-value='50'/>
/_internal/cimg!0/am96m89rxsx
```

opaque-value="percentage" (100)
The transparency value to use, 100 is fully opaque, and 0 is fully transparent.

cs-rgb-hsv="{0, 1}" (0)
Perform rgb to hsv colorspace conversion.

```
<cimg-url src='/internal-roxen-testimage' cs-rgb-hsv='1'/>
/_internal/cimg!0/am96s5d43qb
```

gamma="number" (1.0)
Perform gamma adjustment.

```
<cimg-url src='/internal-roxen-testimage' gamma='0.5'/>
/_internal/cimg!0/amacxg4qaqq
```

```
<cimg-url src='/internal-roxen-testimage' gamma='1.5'/>
/_internal/cimg!0/amacxg4qar1
```

cs-grey="{0, 1}" (0)
Perform rgb to greyscale colorspace conversion.

```
<cimg-url src='/internal-roxen-testimage' cs-grey='1'/>
/_internal/cimg!0/am964gzrg0z
```

cs-invert="{0, 1}" (0)
Invert all colors

```
<cimg-url src='/internal-roxen-testimage' cs-invert='1'/>
/_internal/cimg!0/am9knc6lzhf
```

cs-hsv-rgb="{0, 1}" (0)
Perform hsv to rgb colorspace conversion.

```
<cimg-url src='/internal-roxen-testimage' cs-hsv-rgb='1'/>
/_internal/cimg!0/am9kld5hxib
```

rotate-cw="degree" (0)
Rotate the image clock-wise.

```
<cimg-url src='/internal-roxen-testimage' rotate-cw='20'/>
/_internal/cimg!0/am9kje4dvqe
```

rotate-ccw="degree" (0)
Rotate the image counter clock-wise.

rotate-unit="{rad, deg, ndeg, part}" (deg)
Select the unit to use while rotating.

Unit	Meaning
rad	Radians
deg	Degrees
ndeg	'New' degrees (400 for each full rotation)
part	0 - 1.0 (1.0 == full rotation)

mirror-x="{0, 1}" (0)
Mirror the image around the X-axis.

mirror-y="{0, 1}" (0)
Mirror the image around the Y-axis.

scale="fact" (1.0)
Scale fact times. (0.5 -> half size, 2.0 -> double size)

```
<cimg-url src='/internal-roxen-testimage' scale='0.5'/>
/_internal/cimg!0/amacxg4qaxt
```

scale="x,y"
Scale to the exact size x,y. If either of X or Y is zero, the image is scaled to the specified width or hight, and the value that is zero is scaled in proportion to the other value.

```
<cimg-url src='/internal-roxen-testimage' scale='20,50'/>
/_internal/cimg!0/amacxg4qaxs
```

max-width="xsize"
If width is larger than 'xsize', scale width to 'xsize' while keeping aspect.

max-height="ysize"
If height is larger than 'ysize', scale height to 'ysize' while keeping aspect.

span-width="xsize"
If width is larger than 'xsize', scale width to 'xsize' while keeping aspect. If width is smaller than 'xsize', extend width to 'xsize' by filling the new space with current background color.

```
<cimg-url src='/internal-roxen-testimage' span-width="350" background-color='white'/>
/_internal/cimg!0/am9kra8u3mq
```

span-height="ysize"
If height is larger than 'ysize', scale height to 'ysize' while keeping aspect. If height is smaller than 'ysize', extend height to 'ysize' by filling the new space with current background color.

```
<cimg-url src='/internal-roxen-testimage' span-height="350" background-color='white'/>
/_internal/cimg!0/am9k7jxpjvm
```

x-offset="pixels" (0)
Cut n pixels from the beginning of the X scale.

```
<cimg-url src='/internal-roxen-testimage' x-
```

```
offset='100'/>
  /_internal/cimg!0/am9k5kwlfqd
```

y-offset="pixels" (0)
Cut n pixels from the beginning of the Y scale.

x-size="pixels" (whole image)
Keep n pixels from the beginning of the X scale.

```
  <cimg-url src='/internal-roxen-testimage' x-
size='100'/>
  /_internal/cimg!0/am5a964bd2d
```

y-size="pixels" (whole image)
Keep n pixels from the beginning of the Y scale.

crop="{x1,y1-x2,y2, auto, guides-cross, guides-region}"
Crops the image by using several differen methods. The simplest is to only specify the area to be cropped with x,y-x,y coordinates. By instead selecting "auto" the image will be cropped so that as many pixels as possible with the same color is removed from around the image.

A more advanced cropping method can be used by giving the crop argument "guides-cross". The image will then be cropped around the intersection of two guides inside the image. Guides can be added to e.g. Photoshop and GIMP images. If several guides are present, which ones to use can be selected with the guides-index=x,y attribute, where x and y is the number of the guides. Guides cross cropping is usefull together with max-width and max-height attributes when creating thumb nails.

A combination of guides cross cropping can be used by giving the crop argument "guides-region". In this cropping mode the area enclosed by two horizontal and two vertical guides are saved. Which guides to use is given by the guides-index=x1,y1-x2,y2 attribute, where the x and y parameters are the number of the guides. Guides can also be specified as guides-index=x,y. Then the saved area will be the one enclosed by the buides x,y and x+1,y+1, counting from left and top. Again, combine max-width and max-height makes a good effect, since scaling is performed after cropping.

```
  <cimg-url src='/internal-roxen-
testimage' crop='50,28-150,92'/>
  /_internal/cimg!0/amacxg4q8rr
```

jpeg-quality="percentage" (75)
Set the quality on the output jpeg image.

```
  <cimg-url src='/internal-roxen-
testimage' format='jpeg' jpeg-quality='30'/>
  /_internal/cimg!0/am5a5823941
```

```
  <cimg-url src='/internal-roxen-
testimage' format='jpeg' jpeg-quality='1'/>
  /_internal/cimg!0/am5a5823940
```

jpeg-optimize="{0, 1}" (1)
If 0, do not generate optimal tables. Somewhat faster, but produces bigger files.

jpeg-progressive=="{0, 1}" (0)
Generate progressive jpeg images.

jpeg-smooth="0-100" (0)
Smooth the image while compressing it. This produces smaller files, but might undo the effects of dithering.

bmp-bpp="1,4,8,24" (24)
Force this number of bits per pixel for bmp images.

bmp-windows="{0, 1}" (1)
Windows or OS/2 mode, default is 1. (windows mode)

bmp-rle="{0, 1}" (0)
RLE 'compress' the BMP image.

gd-alpha_index="color" (0)
Color in the colormap to make transparent for GD-images with alpha channel.

pcx-raw="{1, 0}" (0)
If 1, do not RLE encode the PCX image.

pcx-dpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-xdpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-ydpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-xoffset="0-imagexsize-2" (0)
Offset from start of image data to image content for PCX images. Unused by most programs.

pcx-yoffset="0-imageysize-2" (0)
Offset from start of image data to image content for PCX images. Unused by most programs.

tga-raw="{1, 0}" (0)
If 1, do not RLE encode the Targa image.

ps-dpi="0-10000000.0" (75.0)
Dots per inch for the resulting postscript file.

<colorscope></colorscope>

Provided by module: *Tags: RXML 2 tags*

Makes it possible to change the autodetected colors within the tag. Useful when out-of-order parsing occurs, e.g.

```
<define tag="hello">
  <colorscope bgcolor="red">
    <gtext>Hello</gtext>
  </colorscope>
</define>

<table><tr>
  <td bgcolor="red">
    <hello/>
  </td>
</tr></table>
```

It can also successfully be used when the wiretap module is turned off for e.g. performance reasons.

Attributes

text="color"
Set the text color to this value within the scope.

bgcolor="color"
Set the background color to this value within the scope.

link="color"
Set the link color to this value within the scope.

alink="color"
Set the active link color to this value within the scope.

vlink="color"
Set the visited link color to this value within the scope.

<configimage/>

Provided by module: *Tags: RXML 2 tags*

Returns one of the internal Roxen configuration images. The src attribute is required.

Attributes

src="string"
The name of the picture to show.

border="number" (0)
The image border when used as a link.

alt="string" (The src string)
The picture description.

class="string"
This cascading style sheet (CSS) class definition will be applied to the image.
All other attributes will be inherited by the generated img tag.

<diagram></diagram>

Provided by module: *Graphics: Business graphics*

The `<diagram>` tag is used to draw pie, bar, or line charts as well as graphs. It is quite complex with six internal tags.

Attributes

3d="number"
Draws a pie-chart on top of a cylinder, takes the height in pixels of the cylinder as argument.

background="path"
Use an image as background. Valid types are gif-, jpeg- or pnm-images.

bgcolor="color"
Set the background color to use for anti-aliasing.

center="number"
Centers a pie chart around the *n*th slice.

eng
Write numbers in engineering fashion, i.e like 1.2M.

font="font"
Use this font. Can be overridden in the `<legend>`, `<xaxis>`, `<yaxis>` and `<names>` tags.

fontsize="number"
Height of the text in pixels.

height="number"
Height of the diagram in pixels. Will not have effect below 100.

horgrid
Draw a horizontal grid.

labelcolor="color"
Sets the color for the labels of the axis.

legendfontsize="number"
Height of the legend text. *fontsize* is used if this is undefined.

name="string"
Write a name at the top of the diagram.

namecolor="color" (textcolor)
Set the color of the name, by default set by the *textcolor* attribute.

namefont="font"
Set the font for the diagram name.

namesize="number"
Sets the height of the name, by default *fontsize*.

neng
As eng, but 0.1-1.0 is written as 0.xxx.

notrans
Make bgcolor opaque.

rotate="degree"
Rotate a pie chart this much.

textcolor
Set the color for all text.

tonedbox="color1,color2,color3,color4"
Create a background shading between the colors assigned to each of the four corners.

quant="number"
The number of colors that the result image should have. Default is 128 if tonedbox is used and 32 otherwise.

turn
Turn the diagram 90 degrees. Useful when printing large diagrams.

type="{sumbars, normsum, line, bar, pie, graph}"
The type of diagram. This attribute is required.

vertgrid
Draw vertical grid lines.

voidsep="string"
Change the string that means no such value, by default 'VOID'.

width="number"
Set the width of the diagram in pixels. Values below 100 will not take effect. This attribute is required.

xgridspace="number"
Set the space between two vertical grid lines. The unit is the same as for the data.

ygridspace
Set the space between two horizontal grid lines. The unit is the same as for the data.

Regular `<img>` arguments will be passed on to the generated `<img>` tag.

<colors></colors>

Provided by module: *Graphics: Business graphics*

This tag sets the colors for different pie slices, bars or lines. The colors are presented to the tag in a tab separated list.

Attributes

separator="string"
Set the separator between colors, by default tab.

<data></data>

Provided by module: *Graphics: Business graphics*

This tag contains the data the diagram is to visualize It is required that the data is presented to the tag in a tabular or newline separated form.

Attributes

form="{column, row}"
How to interpret the tabular data, by default row.

lineseparator="string"
Use the specified string as lineseparator instead of newline.

noparse
Do not parse the contents by the RXML parser, before data extraction is done.

separator="string"
Set the separator between elements, by default tab.

xnames="number"
If given, treat the first row or column as names for the data to come. If *xnames* is set to a number N, N lines or columns are used. The name will be written along the pie slice or under the bar.

xnamesvert
Write the *xnames* vertically.

<legend></legend>

Provided by module: *Graphics: Business graphics*

A separate legend with description of the different pie slices, bars or lines.The titles are presented to the tag in a tab separated list.

Attributes

separator="string"
Set the separator between legends, by default tab.

<xaxis/>

Provided by module: *Graphics: Business graphics*

Used for specifying the quantity and unit of the x-axis, as well as its scale, in a graph. The `<yaxis>` tag uses the same attributes.

Attributes

start="float"
Limit the start of the diagram at this value. If set to *min* the axis starts at the lowest value in the data.

stop="float"
Limit the end of the diagram at this value.

quantity="string"
Set the name of the quantity of this axis.

unit="string"
Set the name of the unit of this axis.

<xnames></xnames>

Provided by module: *Graphics: Business graphics*

Separate tag that can be used to give names to put along the pie slices or under the bars. The datanames are presented to the tag as a tab separated list. This tag is useful when the diagram is dynamically created. The `<ynames>` tag uses the same attributes.

Attributes

separator="string" (tab)
Set the separator between names, by default tab.

orient="{vert, horiz}"
How to write names, vertically or horizontally.

<yaxis/>

Provided by module: *Graphics: Business graphics*

Used for specifying the quantity and unit of the y-axis, as well as its scale, in a graph or line chart.Se the `<xaxis>` tag for a complete list of attributes.

<ynames></ynames>

Provided by module: *Graphics: Business graphics*

Separate tag that can be used to give names to put along the pie slices or under the bars. The datanames are presented to the tag as a tab separated list. This tag is useful when the diagram is dynamically created. See the `<xnames>` tag for a complete list of attributes.

Some examples:

```
    <diagram type='pie' width='200' height='200'  n
ame='Population'
  tonedbox='lightblue,lightblue,white,white'>
    <data separator=','>5305048,5137269,4399993,886
5051</data>
    <legend separator=','>Denmark,Finland,Norway,Sw
eden</legend>
 </diagram>
    <diagram type='pie' width='200' height='200'  n
ame='Population'
  tonedbox='lightblue,lightblue,white,white'>
    <data separator=','>5305048,5137269,4399993,886
5051</data>
    <legend separator=','>Denmark,Finland,Norway,Sw
eden</legend>
 </diagram>
```

```
   <diagram type='bar' width='200' height='250' nam
e='Population'
```

```
 horgrid='' tonedbox='lightblue,lightblue,white,whi
te'>
   <data xnamesvert='' xnames='' separator=',>
     Denmark,Finland,Norway,Sweden
     5305048,5137269,4399993,8865051
   </data>
 </diagram>
   <diagram type='bar' width='200' height='250' nam
e='Population'
 horgrid='' tonedbox='lightblue,lightblue,white,whi
te'>
   <data xnamesvert='' xnames='' separator=',>
     Denmark,Finland,Norway,Sweden
     5305048,5137269,4399993,8865051
   </data>
 </diagram>



   <diagram type='bar' width='200' height='250'
 name='Age structure' horgrid=''
 tonedbox='lightblue,lightblue,white,white'>
   <data xnamesvert='' xnames='' form='column'
   separator=',>
     Denmark,951175,3556339,797534
     Finland,966593,3424107,746569
     Norway,857952,2846030,696011
     Sweden,1654180,5660410,1550461
   </data>
   <legend separator=',>
     0-14,15-64,65-
   </legend>
 </diagram>
   <diagram type='bar' width='200' height='250'
 name='Age structure' horgrid=''
 tonedbox='lightblue,lightblue,white,white'>
   <data xnamesvert='' xnames='' form='column'
   separator=',>
     Denmark,951175,3556339,797534
     Finland,966593,3424107,746569
     Norway,857952,2846030,696011
     Sweden,1654180,5660410,1550461
   </data>
   <legend separator=',>
     0-14,15-64,65-
   </legend>
 </diagram>



   <diagram type='sumbar' width='200' height='250'
 name='Land Use' horgrid=''
 tonedbox='lightblue,lightblue,white,white'>
   <data xnamesvert='' xnames='' form='column'
   separator=',>
     Denmark,27300,4200,10500
     Finland,24400,231800,48800
     Norway,9240,83160,215600
     Sweden,32880,279480,102750
   </data>
   <legend separator=',>
     Arable,Forests,Other
   </legend>
   <yaxis quantity='area'/>
   <yaxis unit='km^2'/>
 </diagram>
   <diagram type='sumbar' width='200' height='250'
 name='Land Use' horgrid=''
 tonedbox='lightblue,lightblue,white,white'>
   <data xnamesvert='' xnames='' form='column'
   separator=',>
     Denmark,27300,4200,10500
     Finland,24400,231800,48800
     Norway,9240,83160,215600
     Sweden,32880,279480,102750
   </data>
   <legend separator=',>
     Arable,Forests,Other
   </legend>
   <yaxis quantity='area'/>
   <yaxis unit='km^2'/>
 </diagram>



   <diagram type='normsumbar' width='200' height='2
50'
 name='Land Use' horgrid=''
 tonedbox='lightblue,lightblue,white,white'>
   <data xnamesvert='' xnames='' form='column'
   separator=',>
     Denmark,27300,4200,10500
     Finland,24400,231800,48800
     Norway,9240,83160,215600
     Sweden,32880,279480,102750
   </data>
   <legend separator=',>
     Arable,Forests,Other
   </legend>
   <yaxis quantity='%'/>
 </diagram>
   <diagram type='normsumbar' width='200' height='2
50'
 name='Land Use' horgrid=''
 tonedbox='lightblue,lightblue,white,white'>
   <data xnamesvert='' xnames='' form='column'
   separator=',>
     Denmark,27300,4200,10500
     Finland,24400,231800,48800
     Norway,9240,83160,215600
     Sweden,32880,279480,102750
   </data>
   <legend separator=',>
     Arable,Forests,Other
   </legend>
   <yaxis quantity='%'/>
 </diagram>



   <diagram type='line' width='200' height='250'
 name='Exchange Rates' horgrid=''
 tonedbox='lightblue,lightblue,white,white'>
   <data form='row' separator=','
   xnamesvert='' xnames=''>
     1992,1993,1994,1995,1996
     0.166,0.154,0.157,0.179,0.172
     0.223,0.175,0.191,0.229,0.218
     0.161,0.141,0.142,0.158,0.155
     0.172,0.128,0.130,0.149,0.140</data>
   <yaxis start='0.09' stop='0.25'/>
   <legend separator=',>
     Danish kroner (DKr),
     Markkaa (FMk),
     Norwegian kronor (NKr),
     Swedish kronor (SKr)
   </legend>
   <xaxis quantity='year'/>
   <yaxis quantity='US$'/>
 </diagram>
   <diagram type='line' width='200' height='250'
 name='Exchange Rates' horgrid=''
 tonedbox='lightblue,lightblue,white,white'>
   <data form='row' separator=','
   xnamesvert='' xnames=''>
     1992,1993,1994,1995,1996
     0.166,0.154,0.157,0.179,0.172
     0.223,0.175,0.191,0.229,0.218
     0.161,0.141,0.142,0.158,0.155
     0.172,0.128,0.130,0.149,0.140</data>
   <yaxis start='0.09' stop='0.25'/>
   <legend separator=',>
     Danish kroner (DKr),
     Markkaa (FMk),
     Norwegian kronor (NKr),
     Swedish kronor (SKr)
```

```
  </legend>
  <xaxis quantity='year'/>
  <yaxis quantity='US$'/>
</diagram>
```

<gbutton></gbutton>

Provided by module: *Graphics: GButton*

Creates graphical buttons.

Attributes

pagebgcolor="color"

bgcolor="color"
Background color inside and outside button.

```
<gbutton bgcolor='lightblue'>Background</gbutton>
<gbutton bgcolor='lightblue'>Background</gbutton>
```

textcolor="color"
Button text color.

```
<gbutton textcolor='#ff6600'>Text</gbutton>
<gbutton textcolor='#ff6600'>Text</gbutton>
```

frame-image="path"
Use this XCF-image as a frame for the button. The image is required to have at least the following layers: background, mask and frame.

alt="string"
Alternative button and alt text.

href="uri"
Button URI.

textstyle="{normal, condensed}"
Set to *normal* or *condensed* to alter text style.

width=""
Minimum button width.

align="{left, center, right}"
Set text alignment. There are some alignment restrictions: when text alignment is either *left* or *right*, icons must also be aligned *left* or *right*.

state="{enabled, disabled}"
Set to *enabled* or *disabled* to select button state.

icon-src="URI"
Fetch the icon from this URI.

icon-data=""
Inline icon data.

align-icon="{left, center-before, center-after, right}"
Set icon alignment.

left	Place icon on the left side of the text.
center-before	Center the icon before the text. Requires the *align='center'* attribute.
center-after	Center the icon after the text. Requires the *align='center'* attribute.
right	Place icon on the right side of the text.

```
<gbutton width='150' align-icon='center-
before' icon-src='internal-roxen-help'>Roxen 2.0</
gbutton>
<gbutton width='150' align-icon='center-
before' icon-src='internal-roxen-help'>Roxen 2.0</
gbutton>
```

```
<gbutton width='150' align='center' align-
icon='center-after'
icon-src='/internal-roxen-help'>Roxen 2.0</
gbutton>
<gbutton width='150' align='center' align-
icon='center-after'
icon-src='/internal-roxen-help'>Roxen 2.0</
gbutton>
```

valign-icon="{above, middle, below}"
Set icon vertical alignment. Requires three horizontal guidelines in the frame image. If set to *above* the icon is placed between the first and second guidelines and the text between the second and third ones. If set to *below* the placement is reversed. Default value is *middle*.

font="fontname"

format="{gif, jpeg, png, avs, gmp, bd, hrz, ilbm, psx, pnm, ps, pvr, tga, tiff, wbf, xbm, xpm}" (gif)
The format to encode the image to. The formats available are:

Acronym	Acronym interpretation
gif	Graphics Interchange Format (might be missing in your roxen)
jpeg	Joint Photography Expert Group image compression
png	Portable Networks Graphics
avs	Advanced Visual Systems Inc. image format
bmp	Windows BitMaP file
gd	Internal format used by libgd
hrz	HRZ is (was?) used for amatuer radio slow-scan TV.
ilbm	Interchangeable File Format: interleaved bitmap
pcx	Zsoft PCX file format (PC / DOS)

Acronym	Acronym interpretation
pnm	Portable AnyMap
ps	Adobe PostScript file
pvr	Pover VR (dreamcast image)
tga	TrueVision Targa (PC / DOS)
tiff	Tag Image File Format
wbf	WAP Bitmap File
xbm	XWindows Bitmap File
xpm	XWindows Pixmap File

quant="number" (format dependant)
The number of colors to quantizize the image to.
Default for gif is 32(+1 transparent), for most other formats (except black and white) is it unlimited.

dither="{none, random, floyd-steinberg}" (none)
Choose the dithering method.

Method	Meaning
none	No dithering is performed at all.
random	Random scatter dither. Not visually pleasing, but it is useful for very high resolution printing.
floyd-steinberg	Error diffusion dithering. Usually the best dithering method.

true-alpha
If present, render a real alpha channel instead of on/off alpha. If the file format only supports on/off alpha, the alpha channel is dithered using a floyd-steinberg dither.

background-color="color" (taken from the page)
The color to render the image against.

opaque-value="percentage" (100)
The transparency value to use, 100 is fully opaque, and 0 is fully transparent.

cs-rgb-hsv="{0, 1}" (0)
Perform rgb to hsv colorspace conversion.

gamma="number" (1.0)
Perform gamma adjustment.

cs-grey="{0, 1}" (0)
Perform rgb to greyscale colorspace conversion.

cs-invert="{0, 1}" (0)
Invert all colors

cs-hsv-rgb="{0, 1}" (0)
Perform hsv to rgb colorspace conversion.

rotate-cw="degree" (0)
Rotate the image clock-wise.

rotate-ccw="degree" (0)
Rotate the image counter clock-wise.

rotate-unit="{rad, deg, ndeg, part}" (deg)
Select the unit to use while rotating.

Unit	Meaning
rad	Radians
deg	Degrees
ndeg	'New' degrees (400 for each full rotation)
part	0 - 1.0 (1.0 == full rotation)

mirror-x="{0, 1}" (0)
Mirror the image around the X-axis.

mirror-y="{0, 1}" (0)
Mirror the image around the Y-axis.

scale="fact" (1.0)
Scale fact times. (0.5 -> half size, 2.0 -> double size)

scale="x,y"
Scale to the exact size x,y. If either of X or Y is zero, the image is scaled to the specified width or hight, and the value that is zero is scaled in proportion to the other value.

max-width="xsize"
If width is larger than 'xsize', scale width to 'xsize' while keeping aspect.

max-height="ysize"
If height is larger than 'ysize', scale height to 'ysize' while keeping aspect.

span-width="xsize"
If width is larger than 'xsize', scale width to 'xsize' while keeping aspect. If width is smaller than 'xsize', extend width to 'xsize' by filling the new space with current background color.

span-height="ysize"
If height is larger than 'ysize', scale height to 'ysize' while keeping aspect. If height is smaller than 'ysize', extend height to 'ysize' by filling the new space with current background color.

x-offset="pixels" (0)
Cut n pixels from the beginning of the X scale.

y-offset="pixels" (0)
Cut n pixels from the beginning of the Y scale.

x-size="pixels" (whole image)
Keep n pixels from the beginning of the X scale.

y-size="pixels" (whole image)
Keep n pixels from the beginning of the Y scale.

crop="{x1,y1-x2,y2, auto, guides-cross, guides-region}"
Crops the image by using several differen methods. The simplest is to only specify the area to be cropped with x,y-x,y coordinates. By instead selecting "auto" the image will be cropped so that as many pixels as possible with the same color is removed from around the image.
A more advanced cropping method can be used by giving the crop argument "guides-cross". The image will then

be cropped around the intersection of two guides inside the image. Guides can be added to e.g. Photoshop and GIMP images. If several guides are present, which ones to use can be selected with the guides-index=x,y attribute, where x and y is the number of the guides. Guides cross cropping is usefull together with max-width and max-height attributes when creating thumb nails.

A combination of guides cross cropping can be used by giving the crop argument "guides-region". In this cropping mode the area enclosed by two horizontal and two vertical guides are saved. Which guides to use is given by the guides-index=x1,y1-x2,y2 attribute, where the x and y parameters are the number of the guides. Guides can also be specified as guides-index=x,y. Then the saved area will be the one enclosed by the buides x,y and x+1,y+1, counting from left and top. Again, combine max-width and max-height makes a good effect, since scaling is performed after cropping.

jpeg-quality="percentage" (75)
Set the quality on the output jpeg image.

jpeg-optimize="{0, 1}" (1)
If 0, do not generate optimal tables. Somewhat faster, but produces bigger files.

jpeg-progressive=="{0, 1}" (0)
Generate progressive jpeg images.

jpeg-smooth="0-100" (0)
Smooth the image while compressing it. This produces smaller files, but might undo the effects of dithering.

bmp-bpp="1,4,8,24" (24)
Force this number of bits per pixel for bmp images.

bmp-windows="{0, 1}" (1)
Windows or OS/2 mode, default is 1. (windows mode)

bmp-rle="{0, 1}" (0)
RLE 'compress' the BMP image.

gd-alpha_index="color" (0)
Color in the colormap to make transparent for GD-images with alpha channel.

pcx-raw="{1, 0}" (0)
If 1, do not RLE encode the PCX image.

pcx-dpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-xdpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-ydpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-xoffset="0-imagexsize-2" (0)
Offset from start of image data to image content for PCX images. Unused by most programs.

pcx-yoffset="0-imageysize-2" (0)
Offset from start of image data to image content for PCX images. Unused by most programs.

tga-raw="{1, 0}" (0)
If 1, do not RLE encode the Targa image.

ps-dpi="0-10000000.0" (75.0)
Dots per inch for the resulting postscript file.

<gbutton-url></gbutton-url>

Provided by module: *Graphics: GButton*

Generates an URI to the button. `<gbutton-url>` takes the same attributes as *<gbutton>* including the image cache attributes.

Attributes

pagebgcolor="color"

bgcolor="color"
Background color inside and outside button.

```
<gbutton bgcolor='lightblue'>Background</gbutton>
<gbutton bgcolor='lightblue'>Background</gbutton>
```

textcolor="color"
Button text color.

```
<gbutton textcolor='#ff6600'>Text</gbutton>
<gbutton textcolor='#ff6600'>Text</gbutton>
```

frame-image="path"
Use this XCF-image as a frame for the button. The image is required to have at least the following layers: background, mask and frame.

alt="string"
Alternative button and alt text.

href="uri"
Button URI.

textstyle="{normal, condensed}"
Set to *normal* or *condensed* to alter text style.

width=""
Minimum button width.

align="{left, center, right}"
Set text alignment. There are some alignment restrictions: when text alignment is either *left* or *right*, icons must also be aligned *left* or *right*.

state="{enabled, disabled}"
Set to *enabled* or *disabled* to select button state.

icon-src="URI"
Fetch the icon from this URI.

icon-data=""
Inline icon data.

align-icon="{left, center-before, center-after, right}"
Set icon alignment.

left	Place icon on the left side of the text.
center-before	Center the icon before the text. Requires the *align='center'* attribute.
center-after	Center the icon after the text. Requires the *align='center'* attribute.

right	Place icon on the right side of the text.

```
  <gbutton width='150' align-icon='center-
before' icon-src='internal-roxen-help'>Roxen 2.0</
gbutton>
  <gbutton width='150' align-icon='center-
before' icon-src='internal-roxen-help'>Roxen 2.0</
gbutton>
```

```
  <gbutton width='150' align='center' align-
icon='center-after'
  icon-src='/internal-roxen-help'>Roxen 2.0</
gbutton>
  <gbutton width='150' align='center' align-
icon='center-after'
  icon-src='/internal-roxen-help'>Roxen 2.0</
gbutton>
```

valign-icon="{above, middle, below}"
Set icon vertical alignment. Requires three horizontal guidelines in the frame image. If set to *above* the icon is placed between the first and second guidelines and the text between the second and third ones. If set to *below* the placement is reversed. Default value is *middle*.

font="fontname"

format="{gif, jpeg, png, avs, gmp, bd, hrz, ilbm, psx, pnm, ps, pvr, tga, tiff, wbf, xbm, xpm}" (gif)
The format to encode the image to. The formats available are:

Acronym	Acronym interpretation
gif	Graphics Interchange Format (might be missing in your roxen)
jpeg	Joint Photography Expert Group image compression
png	Portable Networks Graphics
avs	Advanced Visual Systems Inc. image format
bmp	Windows BitMaP file
gd	Internal format used by libgd
hrz	HRZ is (was?) used for amatuer radio slow-scan TV.
ilbm	Interchangeable File Format: interleaved bitmap
pcx	Zsoft PCX file format (PC / DOS)
pnm	Portable AnyMap
ps	Adobe PostScript file
pvr	Pover VR (dreamcast image)
tga	TrueVision Targa (PC / DOS)
tiff	Tag Image File Format
wbf	WAP Bitmap File
xbm	XWindows Bitmap File
xpm	XWindows Pixmap File

quant="number" (format dependant)
The number of colors to quantizize the image to.
Default for gif is 32(+1 transparent), for most other formats (except black and white) is it unlimited.

dither="{none, random, floyd-steinberg}" (none)
Choose the dithering method.

Method	Meaning
none	No dithering is performed at all.
random	Random scatter dither. Not visually pleasing, but it is useful for very high resolution printing.
floyd-steinberg	Error diffusion dithering. Usually the best dithering method.

true-alpha
If present, render a real alpha channel instead of on/off alpha. If the file format only supports on/off alpha, the alpha channel is dithered using a floyd-steinberg dither.

background-color="color" (taken from the page)
The color to render the image against.

opaque-value="percentage" (100)
The transparency value to use, 100 is fully opaque, and 0 is fully transparent.

cs-rgb-hsv="{0, 1}" (0)
Perform rgb to hsv colorspace conversion.

gamma="number" (1.0)
Perform gamma adjustment.

cs-grey="{0, 1}" (0)
Perform rgb to greyscale colorspace conversion.

cs-invert="{0, 1}" (0)
Invert all colors

cs-hsv-rgb="{0, 1}" (0)
Perform hsv to rgb colorspace conversion.

rotate-cw="degree" (0)
Rotate the image clock-wise.

rotate-ccw="degree" (0)
Rotate the image counter clock-wise.

rotate-unit="{rad, deg, ndeg, part}" (deg)
Select the unit to use while rotating.

Unit	Meaning
rad	Radians
deg	Degrees
ndeg	'New' degrees (400 for each full rotation)
part	0 - 1.0 (1.0 == full rotation)

mirror-x="{0, 1}" (0)
Mirror the image around the X-axis.

mirror-y="{0, 1}" (0)
Mirror the image around the Y-axis.

scale="fact" (1.0)
Scale fact times. (0.5 -> half size, 2.0 -> double size)

scale="x,y"
Scale to the exact size x,y. If either of X or Y is zero, the image is scaled to the specified width or hight, and the value that is zero is scaled in proportion to the other value.

max-width="xsize"
If width is larger than 'xsize', scale width to 'xsize' while keeping aspect.

max-height="ysize"
If height is larger than 'ysize', scale height to 'ysize' while keeping aspect.

span-width="xsize"
If width is larger than 'xsize', scale width to 'xsize' while keeping aspect. If width is smaller than 'xsize', extend width to 'xsize' by filling the new space with current background color.

span-height="ysize"
If height is larger than 'ysize', scale height to 'ysize' while keeping aspect. If height is smaller than 'ysize', extend height to 'ysize' by filling the new space with current background color.

x-offset="pixels" (0)
Cut n pixels from the beginning of the X scale.

y-offset="pixels" (0)
Cut n pixels from the beginning of the Y scale.

x-size="pixels" (whole image)
Keep n pixels from the beginning of the X scale.

y-size="pixels" (whole image)
Keep n pixels from the beginning of the Y scale.

crop="{x1,y1-x2,y2, auto, guides-cross, guides-region}"
Crops the image by using several differen methods. The simplest is to only specify the area to be cropped with x,y-x,y coordinates. By instead selecting "auto" the image will be cropped so that as many pixels as possible with the same color is removed from around the image.

A more advanced cropping method can be used by giving the crop argument "guides-cross". The image will then be cropped around the intersection of two guides inside the image. Guides can be added to e.g. Photoshop and GIMP images. If several guides are present, which ones to use can be selected with the guides-index=x,y attribute, where x and y is the number of the guides. Guides cross cropping is usefull together with max-width and max-height attributes when creating thumb nails.

A combination of guides cross cropping can be used by giving the crop argument "guides-region". In this cropping mode the area enclosed by two horizontal and two vertical guides are saved. Which guides to use is given by the guides-index=x1,y1-x2,y2 attribute, where the x and y parameters are the number of the guides. Guides can also be specified as guides-index=x,y. Then the saved area will be the one enclosed by the buides x,y and x+1,y+1, counting from left and top. Again, combine max-width and max-height makes a good effect, since scaling is performed after cropping.

jpeg-quality="percentage" (75)
Set the quality on the output jpeg image.

jpeg-optimize="{0, 1}" (1)
If 0, do not generate optimal tables. Somewhat faster, but produces bigger files.

jpeg-progressive==="{0, 1}" (0)
Generate progressive jpeg images.

jpeg-smooth="0-100" (0)
Smooth the image while compressing it. This produces smaller files, but might undo the effects of dithering.

bmp-bpp="1,4,8,24" (24)
Force this number of bits per pixel for bmp images.

bmp-windows="{0, 1}" (1)
Windows or OS/2 mode, default is 1. (windows mode)

bmp-rle="{0, 1}" (0)
RLE 'compress' the BMP image.

gd-alpha_index="color" (0)
Color in the colormap to make transparent for GD-images with alpha channel.

pcx-raw="{1, 0}" (0)
If 1, do not RLE encode the PCX image.

pcx-dpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-xdpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-ydpy="0-10000000.0" (75.0)
Resolution, in pixels per inch.

pcx-xoffset="0-imagexsize-2" (0)
Offset from start of image data to image content for PCX images. Unused by most programs.

pcx-yoffset="0-imageysize-2" (0)
Offset from start of image data to image content for PCX images. Unused by most programs.

tga-raw="{1, 0}" (0)
If 1, do not RLE encode the Targa image.

ps-dpi="0-10000000.0" (75.0)
Dots per inch for the resulting postscript file.

<gh></gh>

Provided by module: *Graphics: Graphic text*

Creates a graphical header. `<gh>` takes the same attributes as *<gtext>*. `<gh>` comes in six flavors, from `<gh1>` through `<gh6>` and are the RXML counterpart to the HTML tags `<h1>` through `<h6>`.

Attributes

verbatim
By default the gtext tag will try to make typographical enhancements to the text to make the resulting image as eye pleasing as possible. If you want to turn this feature off, add this attribute to the tag.

alpha="path"
Use the specified image as an alpha channel, together with the background attribute.

background="path"
Specifies the image to use as background.

tile
Tiles the background and foreground images if they are smaller than the actual image.

mirrortile
Tiles the background and foreground images around x-axis and y-axis for odd frames, creating seamless textures.

bevel="width"
Draws a bevel-box around the text.

bgcolor="color"
Sets the background color. If the module "HTML color wiretap" is loaded, they are taken from the normal HTML tags in your document, like body, table, tr and td.
If you set the background color, it is probably best to add the notrans attribute as well.

bgturbulence="frequency,color;frequency,color..."
Apply a turbulence effect on the background.

bold
Use a bold version of the font, if available. Can not be used together with the black or light attributes.

black
Use a black, or heavy, version of the font, if available. Can not be used together with the bold or light attributes.

light
Use a light version of the font, if available. Can not be used together with the bold or black attributes.

italic
Use an italic version of the font, if available.

bshadow="distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible, but you might have to add 'spacing'.

chisel
Make the text look like it has been cut into the background.

crop
Remove all white-space around the image.

encoding="string"
Choose with which charset the text is encoded with.

fadein="blur,steps,delay,initialdelay"
Generates an animated GIF file of a fade-in effect.

fgcolor="color"
Sets the text color.

font="string"
Selects which font to use. You can get a list of all available fonts by using the list fonts task in the administration interface, or by using the *<emit fonts>* plugin.

fontsize="number"
Selects which size of the font that should be used.

format="string"
Set the image format, e.g. "png".

fs
Apply floyd-steinberg dithering to the resulting image. Most of the time it is much better to increase the number of colors, instead of dithering the image, but sometimes when using very complex background images dithering is O.K.

ghost="dist,blur,color"
Apply a ghost effect. Cannot be used together with shadow or magic coloring.

glow="color"
Apply a 'glow' filter to the image. Quite a CPU eater. Looks much better on a dark background, where a real 'glow' effect can be achieved.

maxlen="number"
Sets the maximum length of the text that will be rendered into an image, by default 300.

move="x,y"
Moves the text relative to the upper left corner of the background image. This will not change the size of the image.

narrow
Use a narroe version of the font, if available.

notrans
Do not make the background transparent. Useful when making 'boxes' of color around the text.

nowhitespace
Removes all whitespaces before and after the real text.

opaque="percentage"
Sets the 'opaque' value of the color used to draw the text. Default is 100%. In the example below, notice how the text color mixes with the two background colors.

outline="color,extra-radius"
Draw an outline around the text. Quite useful when combined with textscale.

pressed
Inverts the direction of the bevel box, to make it look like a button that is pressed down. The magic modifier will do this automatically.

quant="number"
Quantifies the image with this number of colors. Using a lower number will decrease the image (file)size, but make the text look more 'edgy', and if you use complex backgrounds or image textures, more colors will be neded. At

most 255 colors can be used, and less than 2 is quite useless. It is advisable to use powers of 2 to optimize the palette allocation.

rescale
Rescale the background to fill the whole image.

rotate="angle"
Rotates the image this number of degrees counter-clockwise.

scale="number"
Sets the scale of the image. Larger than 1.0 is enlargement.

scolor="color"
Use this color for the shadow. Used with the shadow attribute.

scroll="width,steps,delay"
Generate an animated GIF image of the text scrolling.

shadow="intensity,distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible,

size="width,height"
Set the size of the image.

spacing="number"
Add space around the text.

talign="{left, right, center}"
Adjust the alignment of the text.

textbelow="color"
Place the text centered in a box of the given color below the image area. Useful together with background to make captions for images.

textbox="opaque,color"
Draw a box with an opaque value below the text of the specified color.

textscale="color,color,color,color"
Apply a color filter to the text. The colors are, respectively, upper left, lower left, upper right and lower right. It is probably a good idea to increase the 'quant' value when using this argument.

texture="path"
Uses the specified images as a field texture.

tile
Tiles the background and foreground images if they are smaller than the actual image.

xpad="{percentage, integer ended with "px"}"
Sets the padding between characters. The value can either be an relative change, in percent, or an absolute number of pixels, ended with the string "px". Note that different fonts reacts differently on these values and for some it will not have any effect at all. This depends on the type of the font and the font implementation.

xsize="number"
Sets the width.

xspacing="number"
Sets the horizontal spacing.

ypad="{percentage, integer ended with "px"}"
Sets the padding beteen lines. Works as xpad.

ysize="number"
Sets the height.

yspacing="number"
Sets the vertical spacing.

<gtext></gtext>

Provided by module: *Graphics: Graphic text*

Creates an image with the tag content texts. It is possible to pass attributes, such as the target attribute, to the resulting tags by including them in the gtext tag.

Attributes

alt="string"
Sets the alt attribute of the generated `<img>` tag. By default the alt attribute will be set to the contents of the `<gtext>` tag.

```
    <gtext fgcolor="blue" alt="Hello!">Welcome!</
gtext>
    <gtext fgcolor="blue" alt="Hello!">Welcome!</
gtext>
```

border="width,color"
Draws a border around the text of the specified width and color.

```
  <gtext fgcolor="blue" border="2,red">Red border</
gtext>
  <gtext fgcolor="blue" border="2,red">Red border</
gtext>
```

href="URL"
Link the image to the specified URL. The link color of the document will be used as the default foreground rather than the foreground color.

magic="message"
Used together with the href attribute to generate a JavaScript that will highlight the image when the mouse is moved over it. The message is shown in the browser's status bar.

```
  <gtext href="http://
www.roxen.com" magic="Roxen">www.roxen.com</gtext>
  <gtext href="http://
www.roxen.com" magic="Roxen">www.roxen.com</gtext>
```

magic-attribute="value"
Same as for any `<gtext>` attribute, except for the highlighted image.

```
  <gtext fgcolor="blue" magic=""
magic-glow="red" magic-fg="white">Mouse me!</gtext>
  <gtext fgcolor="blue" magic=""
magic-glow="red" magic-fg="white">Mouse me!</gtext>
```

noxml
Do not terminate the image tag with "/".

split

Make each word into a separate gif image. Useful if you are writing a large text, and word wrap at the edges of the display is desired.

```
   <gtext scale='0.4' split='split'>
 Useful if you are writing a large text, and word w
rap at the edges
 of the display is desired.
 </gtext>
   <gtext scale='0.4' split='split'>
 Useful if you are writing a large text, and word w
rap at the edges
 of the display is desired.
 </gtext>
```

This will allow the browser to word-wrap the text, but will disable certain attributes like *magic*. Note that the word wraping functionality of this example cannot be shown as the size of the browser window is determined by the largest example box.

```
  <gtext scale="0.4" split="">One image per charact
er.</gtext>
  <gtext scale="0.4" split="">One image per charact
er.</gtext>
```

submit
Creates a submit-button for forms. Does not work together with *split* or *magic* attributes.

verbatim
By default the gtext tag will try to make typographical enhancements to the text to make the resulting image as eye pleasing as possible. If you want to turn this feature off, add this attribute to the tag.

alpha="path"
Use the specified image as an alpha channel, together with the background attribute.

background="path"
Specifies the image to use as background.

tile
Tiles the background and foreground images if they are smaller than the actual image.

mirrortile
Tiles the background and foreground images around x-axis and y-axis for odd frames, creating seamless textures.

bevel="width"
Draws a bevel-box around the text.

```
  <gtext bevel="2">Ok</gtext>
  <gtext bevel="2">Ok</gtext>
```

bgcolor="color"
Sets the background color. If the module "HTML color wiretap" is loaded, they are taken from the normal HTML tags in your document, like body, table, tr and td.
If you set the background color, it is probably best to add the notrans attribute as well.

```
  <gtext notrans="" bgcolor="pink">Pink</gtext>
<gtext notrans="" bgcolor="#ff0000">Red</gtext>
<gtext notrans="" bgcolor="%50,0,100,0">%50,0,100,0
</gtext>
  <gtext notrans="" bgcolor="pink">Pink</gtext>
<gtext notrans="" bgcolor="#ff0000">Red</gtext>
<gtext notrans="" bgcolor="%50,0,100,0">%50,0,100,0
</gtext>
```

bgturbulence="frequency,color;frequency,color..."
Apply a turbulence effect on the background.

bold
Use a bold version of the font, if available. Can not be used together with the black or light attributes.

```
  <gtext font='lucida'>Aa3</gtext><br />
<gtext font='lucida' bold=''>Aa3</gtext><br />
<gtext font='lucida' italic=''>Aa3</gtext><br />
<gtext font='lucida' bold='' italic=''>Aa3</
gtext><br />
  <gtext font='lucida'>Aa3</gtext><br />
<gtext font='lucida' bold=''>Aa3</gtext><br />
<gtext font='lucida' italic=''>Aa3</gtext><br />
<gtext font='lucida' bold='' italic=''>Aa3</
gtext><br />
```

black
Use a black, or heavy, version of the font, if available. Can not be used together with the bold or light attributes.

light
Use a light version of the font, if available. Can not be used together with the bold or black attributes.

italic
Use an italic version of the font, if available.

bshadow="distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible, but you might have to add 'spacing'.

```
  <gtext scale="0.8" fgcolor="#FF6600"
bshadow="1">&lt;gtext bshadow=1&gt;</gtext>
<br /><gtext scale="0.8" fgcolor="#FF6600"
bshadow="2">&lt;gtext bshadow=2&gt;</gtext>
  <gtext scale="0.8" fgcolor="#FF6600"
bshadow="1">&lt;gtext bshadow=1&gt;</gtext>
<br /><gtext scale="0.8" fgcolor="#FF6600"
bshadow="2">&lt;gtext bshadow=2&gt;</gtext>
```

chisel
Make the text look like it has been cut into the background.

```
  <gtext font="lucida" bold="" chisel="" talign="ce
nter" tile=""
opaque="70" fgcolor="gold" bevel="2"
background="/internal-roxen-
squares"> Chisel opaque="70"</gtext>
  <gtext font="lucida" bold="" chisel="" talign="ce
nter" tile=""
opaque="70" fgcolor="gold" bevel="2"
background="/internal-roxen-
squares"> Chisel opaque="70"</gtext>
```

crop
Remove all white-space around the image.

encoding="string"
Choose with which charset the text is encoded with.

fadein="blur,steps,delay,initialdelay"

Generates an animated GIF file of a fade-in effect.

fgcolor="color"
Sets the text color.

```
<gtext fgcolor="#0080FF">#0080FF</gtext>
<gtext fgcolor="#0080FF">#0080FF</gtext>
```

font="string"
Selects which font to use. You can get a list of all available fonts by using the list fonts task in the administration interface, or by using the *<emit fonts>* plugin.

fontsize="number"
Selects which size of the font that should be used.

format="string"
Set the image format, e.g. "png".

fs
Apply floyd-steinberg dithering to the resulting image. Most of the time it is much better to increase the number of colors, instead of dithering the image, but sometimes when using very complex background images dithering is O.K.

ghost="dist,blur,color"
Apply a ghost effect. Cannot be used together with shadow or magic coloring.

```
<gtext spacing="2" crop="" ghost="1,1,red">ghost=
1,1,red</gtext>
<gtext spacing="2" crop="" ghost="1,1,red">ghost=
1,1,red</gtext>
```

glow="color"
Apply a 'glow' filter to the image. Quite a CPU eater. Looks much better on a dark background, where a real 'glow' effect can be achieved.

```
<gtext glow="red">&lt;gtext glow=red&gt;</gtext>
<gtext glow="red">&lt;gtext glow=red&gt;</gtext>
```

maxlen="number"
Sets the maximum length of the text that will be rendered into an image, by default 300.

move="x,y"
Moves the text relative to the upper left corner of the background image. This will not change the size of the image.

narrow
Use a narroe version of the font, if available.

notrans
Do not make the background transparent. Useful when making 'boxes' of color around the text.

```
<gtext bgcolor="red">&lt;gtext bgcolor=red&gt;</
gtext><br />
<gtext bgcolor="red" notrans="">&lt;gtext
bgcolor=red notrans&gt;</gtext>
<gtext bgcolor="red">&lt;gtext bgcolor=red&gt;</
gtext><br />
<gtext bgcolor="red" notrans="">&lt;gtext
bgcolor=red notrans&gt;</gtext>
```

nowhitespace
Removes all whitespaces before and after the real text.

opaque="percentage"
Sets the 'opaque' value of the color used to draw the text. Default is 100%. In the example below, notice how the text color mixes with the two background colors.

```
<gtext scale="0.6" textbox="100,pink,-
11" bgcolor="lightblue"
notrans="" opaque="40" fgcolor="black"
>&lt;Demonstration of opaque text&gt;</gtext>
<gtext scale="0.6" textbox="100,pink,-
11" bgcolor="lightblue"
notrans="" opaque="40" fgcolor="black"
>&lt;Demonstration of opaque text&gt;</gtext>
```

outline="color,extra-radius"
Draw an outline around the text. Quite useful when combined with textscale.

```
<gtext xspacing="4" quant="128" textscale="red,re
d,yellow,yellow"
outline="black,1"
>black, 2 pixels</gtext>
<gtext xspacing="4" quant="128" textscale="red,re
d,yellow,yellow"
outline="black,1"
>black, 2 pixels</gtext>
```

pressed
Inverts the direction of the bevel box, to make it look like a button that is pressed down. The magic modifier will do this automatically.

quant="number"
Quantifies the image with this number of colors. Using a lower number will decrease the image (file)size, but make the text look more 'edgy', and if you use complex backgrounds or image textures, more colors will be neded. At most 255 colors can be used, and less than 2 is quite useless. It is advisable to use powers of 2 to optimize the palette allocation.

```
<gtext quant="2">A</gtext>
<gtext quant="6">A</gtext>
<gtext quant="20">A</gtext>
<gtext quant="200">A</gtext>
<gtext quant="2">A</gtext>
<gtext quant="6">A</gtext>
<gtext quant="20">A</gtext>
<gtext quant="200">A</gtext>
```

rescale
Rescale the background to fill the whole image.

rotate="angle"
Rotates the image this number of degrees counterclockwise.

scale="number"
Sets the scale of the image. Larger than 1.0 is enlargement.

```
<gtext scale="1.0">&lt;gtext scale=1.0&gt;</
gtext>
<gtext scale="0.5">&lt;gtext scale=0.5&gt;</gtext>
<gtext scale="1.0">&lt;gtext scale=1.0&gt;</
gtext>
```

```
<gtext scale="0.5">&lt;gtext scale=0.5&gt;</gtext>
```

scolor="color"
Use this color for the shadow. Used with the shadow attribute.

scroll="width,steps,delay"
Generate an animated GIF image of the text scrolling.

shadow="intensity,distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible,

```
  <gtext scale="0.8" fgcolor="blue"
shadow="40,0">&lt;gtext shadow=40,0&gt;</gtext>
<br /><gtext scale="0.8" fgcolor="blue"
shadow="40,2">&lt;gtext shadow=40,2&gt;</gtext>
  <gtext scale="0.8" fgcolor="blue"
shadow="40,0">&lt;gtext shadow=40,0&gt;</gtext>
<br /><gtext scale="0.8" fgcolor="blue"
shadow="40,2">&lt;gtext shadow=40,2&gt;</gtext>
```

size="width,height"
Set the size of the image.

spacing="number"
Add space around the text.

talign="{left, right, center}"
Adjust the alignment of the text.

textbelow="color"
Place the text centered in a box of the given color below the image area. Useful together with background to make captions for images.

```
  <img src="/internal-roxen-roxen" />  
<gtext scale="0.5" background="/internal-roxen-
roxen"
textbelow="#c0c0c0">Roxen</gtext>
  <img src="/internal-roxen-roxen" />  
<gtext scale="0.5" background="/internal-roxen-
roxen"
textbelow="#c0c0c0">Roxen</gtext>
```

textbox="opaque,color"
Draw a box with an opaque value below the text of the specified color.

textscale="color,color,color,color"
Apply a color filter to the text. The colors are, respectively, upper left, lower left, upper right and lower right. It is probably a good idea to increase the 'quant' value when using this argument.

```
  <gtext quant="128" textscale="blue,white,red,dark
green"
>Blue, white,
red, darkgreen</gtext>
  <gtext quant="128" textscale="blue,white,red,dark
green"
>Blue, white,
red, darkgreen</gtext>
```

texture="path"
Uses the specified images as a field texture.

```
  <gtext texture="/internal-roxen-squares"
 tile="1" fontsize="100">A</gtext>
  <gtext texture="/internal-roxen-squares"
 tile="1" fontsize="100">A</gtext>
```

tile
Tiles the background and foreground images if they are smaller than the actual image.

xpad="{percentage, integer ended with "px"}"
Sets the padding between characters. The value can either be an relative change, in percent, or an absolute number of pixels, ended with the string "px". Note that different fonts reacts differently on these values and for some it will not have any effect at all. This depends on the type of the font and the font implementation.

```
  <gtext font="niquel">HELLO ROXEN</gtext><br />
<gtext xpad="4px" font="niquel">HELLO ROXEN</
gtext><br />
<gtext xpad="50%" font="niquel">HELLO ROXEN</
gtext><br />
  <gtext font="niquel">HELLO ROXEN</gtext><br />
<gtext xpad="4px" font="niquel">HELLO ROXEN</
gtext><br />
<gtext xpad="50%" font="niquel">HELLO ROXEN</
gtext><br />
```

xsize="number"
Sets the width.

xspacing="number"
Sets the horizontal spacing.

ypad="{percentage, integer ended with "px"}"
Sets the padding beteen lines. Works as xpad.

ysize="number"
Sets the height.

yspacing="number"
Sets the vertical spacing.

<gtext-id/>

Provided by module: *Graphics: Graphic text*

Returns an internal URL to an image with the specified gtext attributes applied. Rendering a text image with the given arguments is accomplished by concatenating the text of your choice to the end of the URL returned by the tag, as in "`<gtext-id/>`Hello%20World!".

Be aware that this tag could be abused for denial of service attacks by malicious users swarming your server with requests for images of great length that the server would happily try to render for them. Hence this tag should only be used in environments where you trust all your users, e.g. Intranets.

In most cases the tag `<gtext-url>` solves this problem, but if you would like to generate new text images live without reloading some RXML page, you need this tag. An example application:

```
<define variable='var.id' preparse='' trimwhites=''
>
<gtext-id font='FranklinGothicDemi' fgcolor='blue'/
>
</define>
```

```
<img src='&var.id;Please type some text here:'
     alt='' name='banner' width='468' height='60' /
>

<script language='javascript'>
var image = document.images.banner;
function alter_image(label)
{
  image.src = '&var.id:js;' + escape(label.value);
  label.focus();
  return false;
}
</script>

<form onsubmit='return alter_image(this.label);'>
<input type='text' size='40' name='label' />
</form>
```

Attributes

short
Returns a relative path to the image, i.e. a shorter one.

verbatim
By default the gtext tag will try to make typographical enhancements to the text to make the resulting image as eye pleasing as possible. If you want to turn this feature off, add this attribute to the tag.

alpha="path"
Use the specified image as an alpha channel, together with the background attribute.

background="path"
Specifies the image to use as background.

tile
Tiles the background and foreground images if they are smaller than the actual image.

mirrortile
Tiles the background and foreground images around x-axis and y-axis for odd frames, creating seamless textures.

bevel="width"
Draws a bevel-box around the text.

```
  <gtext bevel="2">Ok</gtext>
  <gtext bevel="2">Ok</gtext>
```

bgcolor="color"
Sets the background color. If the module "HTML color wiretap" is loaded, they are taken from the normal HTML tags in your document, like body, table, tr and td.
If you set the background color, it is probably best to add the notrans attribute as well.

```
  <gtext notrans="" bgcolor="pink">Pink</gtext>
<gtext notrans="" bgcolor="#ff0000">Red</gtext>
<gtext notrans="" bgcolor="%50,0,100,0">%50,0,100,0
</gtext>
  <gtext notrans="" bgcolor="pink">Pink</gtext>
<gtext notrans="" bgcolor="#ff0000">Red</gtext>
<gtext notrans="" bgcolor="%50,0,100,0">%50,0,100,0
</gtext>
```

bgturbulence="frequency,color;frequency,color..."
Apply a turbulence effect on the background.

bold
Use a bold version of the font, if available. Can not be used together with the black or light attributes.

```
  <gtext font='lucida'>Aa3</gtext><br />
<gtext font='lucida' bold=''>Aa3</gtext><br />
<gtext font='lucida' italic=''>Aa3</gtext><br />
<gtext font='lucida' bold='' italic=''>Aa3</
gtext><br />
  <gtext font='lucida'>Aa3</gtext><br />
<gtext font='lucida' bold=''>Aa3</gtext><br />
<gtext font='lucida' italic=''>Aa3</gtext><br />
<gtext font='lucida' bold='' italic=''>Aa3</
gtext><br />
```

black
Use a black, or heavy, version of the font, if available. Can not be used together with the bold or light attributes.

light
Use a light version of the font, if available. Can not be used together with the bold or black attributes.

italic
Use an italic version of the font, if available.

bshadow="distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible, but you might have to add 'spacing'.

```
  <gtext scale="0.8" fgcolor="#FF6600"
bshadow="1">&lt;gtext bshadow=1&gt;</gtext>
<br /><gtext scale="0.8" fgcolor="#FF6600"
bshadow="2">&lt;gtext bshadow=2&gt;</gtext>
  <gtext scale="0.8" fgcolor="#FF6600"
bshadow="1">&lt;gtext bshadow=1&gt;</gtext>
<br /><gtext scale="0.8" fgcolor="#FF6600"
bshadow="2">&lt;gtext bshadow=2&gt;</gtext>
```

chisel
Make the text look like it has been cut into the background.

```
  <gtext font="lucida" bold="" chisel="" talign="ce
nter" tile=""
opaque="70" fgcolor="gold" bevel="2"
background="/internal-roxen-
squares"> Chisel opaque="70"</gtext>
  <gtext font="lucida" bold="" chisel="" talign="ce
nter" tile=""
opaque="70" fgcolor="gold" bevel="2"
background="/internal-roxen-
squares"> Chisel opaque="70"</gtext>
```

crop
Remove all white-space around the image.

encoding="string"
Choose with which charset the text is encoded with.

fadein="blur,steps,delay,initialdelay"
Generates an animated GIF file of a fade-in effect.

fgcolor="color"
Sets the text color.

```
  <gtext fgcolor="#0080FF">#0080FF</gtext>
  <gtext fgcolor="#0080FF">#0080FF</gtext>
```

font="string"
Selects which font to use. You can get a list of all available fonts by using the list fonts task in the administration interface, or by using the *<emit fonts>* plugin.

fontsize="number"
Selects which size of the font that should be used.

format="string"
Set the image format, e.g. "png".

fs
Apply floyd-steinberg dithering to the resulting image. Most of the time it is much better to increase the number of colors, instead of dithering the image, but sometimes when using very complex background images dithering is O.K.

ghost="dist,blur,color"
Apply a ghost effect. Cannot be used together with shadow or magic coloring.

```
  <gtext spacing="2" crop="" ghost="1,1,red">ghost=
1,1,red</gtext>
  <gtext spacing="2" crop="" ghost="1,1,red">ghost=
1,1,red</gtext>
```

glow="color"
Apply a 'glow' filter to the image. Quite a CPU eater. Looks much better on a dark background, where a real 'glow' effect can be achieved.

```
  <gtext glow="red">&lt;gtext glow=red&gt;</gtext>
  <gtext glow="red">&lt;gtext glow=red&gt;</gtext>
```

maxlen="number"
Sets the maximum length of the text that will be rendered into an image, by default 300.

move="x,y"
Moves the text relative to the upper left corner of the background image. This will not change the size of the image.

narrow
Use a narroe version of the font, if available.

notrans
Do not make the background transparent. Useful when making 'boxes' of color around the text.

```
  <gtext bgcolor="red">&lt;gtext bgcolor=red&gt;</
gtext><br />
<gtext bgcolor="red" notrans="">&lt;gtext
bgcolor=red notrans&gt;</gtext>
  <gtext bgcolor="red">&lt;gtext bgcolor=red&gt;</
gtext><br />
<gtext bgcolor="red" notrans="">&lt;gtext
bgcolor=red notrans&gt;</gtext>
```

nowhitespace
Removes all whitespaces before and after the real text.

opaque="percentage"
Sets the 'opaque' value of the color used to draw the text. Default is 100%. In the example below, notice how the text color mixes with the two background colors.

```
  <gtext scale="0.6" textbox="100,pink,-
11" bgcolor="lightblue"
notrans="" opaque="40" fgcolor="black"
>&lt;Demonstration of opaque text&gt;</gtext>
  <gtext scale="0.6" textbox="100,pink,-
11" bgcolor="lightblue"
notrans="" opaque="40" fgcolor="black"
>&lt;Demonstration of opaque text&gt;</gtext>
```

outline="color,extra-radius"
Draw an outline around the text. Quite useful when combined with textscale.

```
  <gtext xspacing="4" quant="128" textscale="red,re
d,yellow,yellow"
outline="black,1"
>black, 2 pixels</gtext>
  <gtext xspacing="4" quant="128" textscale="red,re
d,yellow,yellow"
outline="black,1"
>black, 2 pixels</gtext>
```

pressed
Inverts the direction of the bevel box, to make it look like a button that is pressed down. The magic modifier will do this automatically.

quant="number"
Quantifies the image with this number of colors. Using a lower number will decrease the image (file)size, but make the text look more 'edgy', and if you use complex backgrounds or image textures, more colors will be neded. At most 255 colors can be used, and less than 2 is quite useless. It is advisable to use powers of 2 to optimize the palette allocation.

```
  <gtext quant="2">A</gtext>
<gtext quant="6">A</gtext>
<gtext quant="20">A</gtext>
<gtext quant="200">A</gtext>
  <gtext quant="2">A</gtext>
<gtext quant="6">A</gtext>
<gtext quant="20">A</gtext>
<gtext quant="200">A</gtext>
```

rescale
Rescale the background to fill the whole image.

rotate="angle"
Rotates the image this number of degrees counter-clockwise.

scale="number"
Sets the scale of the image. Larger than 1.0 is enlargement.

```
  <gtext scale="1.0">&lt;gtext scale=1.0&gt;</
gtext>
<gtext scale="0.5">&lt;gtext scale=0.5&gt;</gtext>
  <gtext scale="1.0">&lt;gtext scale=1.0&gt;</
gtext>
<gtext scale="0.5">&lt;gtext scale=0.5&gt;</gtext>
```

scolor="color"
Use this color for the shadow. Used with the shadow attribute.

scroll="width,steps,delay"
Generate an animated GIF image of the text scrolling.

shadow="intensity,distance"

Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible,

```
  <gtext scale="0.8" fgcolor="blue"
shadow="40,0">&lt;gtext shadow=40,0&gt;</gtext>
<br /><gtext scale="0.8" fgcolor="blue"
shadow="40,2">&lt;gtext shadow=40,2&gt;</gtext>
  <gtext scale="0.8" fgcolor="blue"
shadow="40,0">&lt;gtext shadow=40,0&gt;</gtext>
<br /><gtext scale="0.8" fgcolor="blue"
shadow="40,2">&lt;gtext shadow=40,2&gt;</gtext>
```

size="width,height"
Set the size of the image.

spacing="number"
Add space around the text.

talign="{left, right, center}"
Adjust the alignment of the text.

textbelow="color"
Place the text centered in a box of the given color below the image area. Useful together with background to make captions for images.

```
  <img src="/internal-roxen-roxen" />  
<gtext scale="0.5" background="/internal-roxen-
roxen"
textbelow="#c0c0c0">Roxen</gtext>
  <img src="/internal-roxen-roxen" />  
<gtext scale="0.5" background="/internal-roxen-
roxen"
textbelow="#c0c0c0">Roxen</gtext>
```

textbox="opaque,color"
Draw a box with an opaque value below the text of the specified color.

textscale="color,color,color,color"
Apply a color filter to the text. The colors are, respectively, upper left, lower left, upper right and lower right. It is probably a good idea to increase the 'quant' value when using this argument.

```
  <gtext quant="128" textscale="blue,white,red,dark
green"
>Blue, white,
red, darkgreen</gtext>
  <gtext quant="128" textscale="blue,white,red,dark
green"
>Blue, white,
red, darkgreen</gtext>
```

texture="path"
Uses the specified images as a field texture.

```
  <gtext texture="/internal-roxen-squares"
 tile="1" fontsize="100">A</gtext>
  <gtext texture="/internal-roxen-squares"
 tile="1" fontsize="100">A</gtext>
```

tile
Tiles the background and foreground images if they are smaller than the actual image.

xpad="{percentage, integer ended with "px"}"
Sets the padding between characters. The value can either be an relative change, in percent, or an absolute number of pixels, ended with the string "px". Note that different fonts reacts differently on these values and for some it will not have any effect at all. This depends on the type of the font and the font implementation.

```
  <gtext font="niquel">HELLO ROXEN</gtext><br />
<gtext xpad="4px" font="niquel">HELLO ROXEN</
gtext><br />
<gtext xpad="50%" font="niquel">HELLO ROXEN</
gtext><br />
  <gtext font="niquel">HELLO ROXEN</gtext><br />
<gtext xpad="4px" font="niquel">HELLO ROXEN</
gtext><br />
<gtext xpad="50%" font="niquel">HELLO ROXEN</
gtext><br />
```

xsize="number"
Sets the width.

xspacing="number"
Sets the horizontal spacing.

ypad="{percentage, integer ended with "px"}"
Sets the padding beteen lines. Works as xpad.

ysize="number"
Sets the height.

yspacing="number"
Sets the vertical spacing.

<gtext-url></gtext-url>

Provided by module: *Graphics: Graphic text*

Returns an internal URL to an image with the specified attributes applied.

Attributes

short
Returns a relative path to the image, i.e. a shorter one.

verbatim
By default the gtext tag will try to make typographical enhancements to the text to make the resulting image as eye pleasing as possible. If you want to turn this feature off, add this attribute to the tag.

alpha="path"
Use the specified image as an alpha channel, together with the background attribute.

background="path"
Specifies the image to use as background.

tile
Tiles the background and foreground images if they are smaller than the actual image.

mirrortile
Tiles the background and foreground images around x-axis and y-axis for odd frames, creating seamless textures.

bevel="width"
Draws a bevel-box around the text.

bgcolor="color"

Sets the background color. If the module "HTML color wiretap" is loaded, they are taken from the normal HTML tags in your document, like body, table, tr and td.

If you set the background color, it is probably best to add the notrans attribute as well.

bgturbulence="frequency,color;frequency,color..."
Apply a turbulence effect on the background.

bold
Use a bold version of the font, if available. Can not be used together with the black or light attributes.

black
Use a black, or heavy, version of the font, if available. Can not be used together with the bold or light attributes.

light
Use a light version of the font, if available. Can not be used together with the bold or black attributes.

italic
Use an italic version of the font, if available.

bshadow="distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible, but you might have to add 'spacing'.

chisel
Make the text look like it has been cut into the background.

crop
Remove all white-space around the image.

encoding="string"
Choose with which charset the text is encoded with.

fadein="blur,steps,delay,initialdelay"
Generates an animated GIF file of a fade-in effect.

fgcolor="color"
Sets the text color.

font="string"
Selects which font to use. You can get a list of all available fonts by using the list fonts task in the administration interface, or by using the <*emit fonts*> plugin.

fontsize="number"
Selects which size of the font that should be used.

format="string"
Set the image format, e.g. "png".

fs
Apply floyd-steinberg dithering to the resulting image. Most of the time it is much better to increase the number of colors, instead of dithering the image, but sometimes when using very complex background images dithering is O.K.

ghost="dist,blur,color"
Apply a ghost effect. Cannot be used together with shadow or magic coloring.

glow="color"
Apply a 'glow' filter to the image. Quite a CPU eater. Looks much better on a dark background, where a real 'glow' effect can be achieved.

maxlen="number"
Sets the maximum length of the text that will be rendered into an image, by default 300.

move="x,y"
Moves the text relative to the upper left corner of the background image. This will not change the size of the image.

narrow
Use a narroe version of the font, if available.

notrans
Do not make the background transparent. Useful when making 'boxes' of color around the text.

nowhitespace
Removes all whitespaces before and after the real text.

opaque="percentage"
Sets the 'opaque' value of the color used to draw the text. Default is 100%. In the example below, notice how the text color mixes with the two background colors.

outline="color,extra-radius"
Draw an outline around the text. Quite useful when combined with textscale.

pressed
Inverts the direction of the bevel box, to make it look like a button that is pressed down. The magic modifier will do this automatically.

quant="number"
Quantifies the image with this number of colors. Using a lower number will decrease the image (file)size, but make the text look more 'edgy', and if you use complex backgrounds or image textures, more colors will be neded. At most 255 colors can be used, and less than 2 is quite useless. It is advisable to use powers of 2 to optimize the palette allocation.

rescale
Rescale the background to fill the whole image.

rotate="angle"
Rotates the image this number of degrees counter-clockwise.

scale="number"
Sets the scale of the image. Larger than 1.0 is enlargement.

scolor="color"
Use this color for the shadow. Used with the shadow attribute.

scroll="width,steps,delay"
Generate an animated GIF image of the text scrolling.

shadow="intensity,distance"
Draw a blured black drop-shadow behind the text. Using 0 as distance does not currently place the shadow directly below the text. Using negative values for distance is possible,

size="width,height"
Set the size of the image.

spacing="number"
Add space around the text.

talign="{left, right, center}"
Adjust the alignment of the text.

textbelow="color"
Place the text centered in a box of the given color below the image area. Useful together with background to make captions for images.

textbox="opaque,color"
Draw a box with an opaque value below the text of the specified color.

textscale="color,color,color,color"
Apply a color filter to the text. The colors are, respectively, upper left, lower left, upper right and lower right. It is probably a good idea to increase the 'quant' value when using this argument.

texture="path"
Uses the specified images as a field texture.

tile
Tiles the background and foreground images if they are smaller than the actual image.

xpad="{percentage, integer ended with "px"}"
Sets the padding between characters. The value can either be an relative change, in percent, or an absolute number of pixels, ended with the string "px". Note that different fonts reacts differently on these values and for some it will not have any effect at all. This depends on the type of the font and the font implementation.

xsize="number"
Sets the width.

xspacing="number"
Sets the horizontal spacing.

ypad="{percentage, integer ended with "px"}"
Sets the padding beteen lines. Works as xpad.

ysize="number"
Sets the height.

yspacing="number"
Sets the vertical spacing.

<imgs/>

Provided by module: *Tags: RXML 2 tags*

Generates a image tag with the correct dimensions in the width and height attributes. These dimensions are read from the image itself, so the image must exist when the tag is generated. The image must also be in GIF, JPEG/JFIF or PNG format.

Attributes

src="string"
The path to the file that should be shown.

alt="string"
Description of the image. If no description is provided, the filename (capitalized, without extension and with some characters replaced) will be used.

All other attributes will be inherited by the generated img tag.

<tablist></tablist>

Provided by module: *Graphics: Tab list*

`<tablist>` produces graphical navigationtabs. For example, the Administration interface for Roxen WebServer uses tablists for easier administration.

The `<tablist>` tag is by design a wrapper for the *<gbutton>* tag, i.e. it inherits all `<gbutton>` attributes. Also, the `<tab>` tag is in turn a wrapper for `<tablist>` meaning that all attributes which may be given to `<tablist>` may also be used in `<tab>`. Those attributes given to `<tablist>` has a global effect on the tablist, while the same attributes given to a `<tab>` only will have a local effect, thus overriding the globally given attribute.

All contents inside `<tablist>` except for the `<tab>` tags will be discarded. `<taglist>` is used in this way to make it possible for tabs to look different when they are for instance first or last in the tablisting.

The *<define>* tag can be used to globally define the tablist *fgcolor* (foreground color). The define, *<define name="fgcolor">*, declared prior to the `<tablist>` tag, will be sent as an extra argument to `<gbutton>`.

```
  <tablist>
<tab selected='selected'>Information</tab>
<tab>Settings</tab>
</tablist>
  <tablist>
<tab selected='selected'>Information</tab>
<tab>Settings</tab>
</tablist>
```

Attributes

frame-image="" (/internal-roxen-tabframe)
A layered Photoshop (PSD) or Gimp (XCF) image which portrays the tab's appearance. Descriptions of the different layers follows below. If a `<define name="frame-image">`Image_path`</define>` definition is set that image will be the default value instead of `/internal-roxen-tab-frame`.

selcolor="color" (white)
This attribute sets the backgroundcolor for the image. The effect of this attribute is only shown when the attribute "selected" has been set. If a `<define name="selcolor">`colordefinition`</define>` definition is set that color will be the default value instead of `white`.

seltextcolor="color" (black)
This attribute sets the textcolor for the image. The effect of this attribute is only shown when the attribute "selected" has been set. If a `<define name="seltext-color">`colordefinition`</define>` definition is set that color will be the default value instead of `black`. If this definition is not present, the attribute "textcolor", the definition "textcolor" and finally the color "black" will be tested.

dimcolor="color" (#003366)
This attribute sets the backgroundcolor for the image. The effect of this attribute is only shown when the attribute "selected" has *not* been set. If a `<define name="dim-color">`colordefinition`</define>` definition is set that color will be the default value instead of `#003366` .

textcolor="color" (white)

This attribute sets the textcolor for the image. The effect of this attribute is only shown when the attribute "selected" has *not* been set. If a `<define name="textcolor">`colordefinition`</define>` definition is set that color will be the default value instead of `white` .

<tab></tab>

Provided by module: *Graphics: Tab list*

`<tab>` defines the layout and function for each and one of the tabs in the tablisting. `<tab>` inherits all attributes available to `<tablist>`, hence all attributes available to *<gbutton>* tag may be used with the `<tab>` tag. For instance, the attribute *href* is very useful when using `<tab>` and a part of *<gbutton>*. For more information about `<gbutton>` attributes, see its documentation.

The contents of the `<tab>` is the tabs text.

Below follows a listing of the attributes unique to the `<tab>` tag. Also, a listing of how imagelayers may be used is presented.

Attributes

selected=""
Using this attribute the layer "selected" in the image will be shown in the generated image. If this attribute has not been given the layer "unselected" will be shown in the generated image.

alt="text" (the tags content)
This attribute sets the alt-text for the tab. By default the alt-text is fetched from the content between the `<tab>`...`</tab>`.

Image Layers

These lists shows the function of the different image layers as well as how one layer from each group may be combined.

Layer Position
Position layers are the layername prefix.

first
A layer with this prefix is only shown for the *first*`<tab>` tag inside the `<tablist>` tag.

last
A layer with this prefix is only shown for the *last*`<tab>` tag inside the `<tablist>` tag.

Layer Focus
Focus layers are the middle part of the layername.

selected
This layer is only shown when the attribute *selected* has been set.

unselected
This layer is only shown when the attribute *selected* has *not* been set.

Layer Type
Type layers are the layername suffix.

[nothing, i.e. ""]
This layer is inserted above all layers in the image, closest to the viewer that is, if lower layers are considered to further in inside the monitor.

mask
This layer should be transparent where the tab is supposed to be transparent. The only thing that is retrieved from this layer is the mask; any graphical content here will be thrown away.

frame
The framelayer contains the various graphical elements fromwhich the frame around the button is built. This layer will always be run in "Multiply" mode, regardless of what mode it was previously set to. "Multiply" adjusts the framelayers brightness, i.e. Value ("V" in HSV), without affecting the colorcomponents, i.e. Hue and Saturation ("HS" in HSV).

background
This layer will be put beneath the *frame* layer and the printed text.

left
This layer is put on the left side of the of the generated image, thus increasing the width of the images left side.

right
This layer is put on the right side of the of the generated image, thus increasing the width of the images right side.

above
This layer will be shown above the other parts of the generated image, thus increasing the height of the images top.

below
This layer will be shown below the other parts of the generated image, thus increasing the height of the images base.

Handling layers

The *Position*- and *Focus*-layers give instructions on *when* the layer is used while the *Type*-layers indicates its *function*.

Position	Focus	Type
""	""	""
first	selected	background
last	unselected	mask
		frame
		left
		right
		above
		below

These three layertypes can be combined into all possible permutations. The order in the name is always *Position Focus Type*, each type separated by a space. If one or two

of the three layertypes is left out, the layer will be shown regardless the extra criterias that might be choosen. For instance, "selected frame" will be shown for the "first" and "last" tabs as well as for the ones in between the two, given that the tab has been marked as "selected".

None of these layers are strictly necessary, as long as there exists at least one layer of the type "background" or "frame". If all "mask"-layers are left out, the mask will primary be the framelayer and secondly the background-layer, if the framelayer is not available.

Emit Tags

This chapter discusses the use of *<emit>* and its various sources.

<emit></emit>

Provided by module: *Tags: RXML 2 tags*

`<emit>` is a generic tag used to fetch data from a provided source, loop over it and assign it to RXML variables accessible through entities.

Occasionally an `<emit>` operation fails to produce output. This might happen when `<emit>` can't find any matches or if the developer has made an error. When this happens the truth value of that page is set to *false*. By using *<else>* afterwards it's possible to detect when an `<emit>` operation fails.

Attributes

source="plugin"
The source from which the data should be fetched.

scope="name" (The emit source)
The name of the scope within the emit tag.

maxrows="number"
Limits the number of rows to this maximum. Note that it is often better to restrict the number of rows emitted by modifying the arguments to the emit plugin, if possible. E.g. when quering a MySQL database the data can be restricted to the first 100 entries by adding "LIMIT 100".

skiprows="number"
Makes it possible to skip the first rows of the result. Negative numbers means to skip everything execept the last n rows. Note that it is often better to make the plugin skip initial rows, if possible.

rowinfo="variable"
The number of rows in the result, after it has been filtered and limited by maxrows and skiprows, will be put in this variable, if given. Note that this may not be the same value as the number of emit iterations that the emit tag will perform, since it will always make one iteration when the attribute do-once is set.

remainderinfo="variable"
The number of rows left to output, when the emit is restricted by the maxrows attribute. Rows excluded by other means such as skiprows or filter are not included in hte remainderinfo value. The rows counted in the remainderinfo are also filtered if the filter attribute is used, so the value represents the actual number of rows that should have been outputed, had emit not been restriced.

do-once
Indicate that at least one loop should be made. All variables in the emit scope will be empty, except for the counter variable.

filter="list"
The filter attribute is used to block certain 'rows' from the source from being emitted. The filter attribute should be set to a list with variable names in the emitted scope and glob patterns the variable value must match in order to not get filtered. A list might look like `name=a*,id=??3?45`. Note that it is often better to perform the filtering in the plugin, by modifying its arguments, if possible. E.g. when querying an SQL database the use of where statements is recommended, e.g. "WHERE name LIKE 'a%' AND id LIKE '__3_45'" will perform the same filtering as above.

```
  <emit source='values' values='foo,bar,baz' split=
',' filter='value=b*'>
&_.value;
</emit>

bar

baz
```

sort="list"
The emit result can be sorted by the emit tag before being output. Just list the variable names in the scope that the result should be sorted on, in prioritized order, e.g. "lastname,firstname". By adding a "-" sign in front of a name, that entry will be sorted in the reversed order. The sort algorithm will treat numbers as complete numbers and not digits in a string, hence "foo8bar" will be sorted before "foo11bar".

&_.counter;
Provided by *Tags: RXML 2 tags*
Gives the current number of loops inside the `<emit>` tag.

<emit source="atlas"></emit>

Provided by module: *Graphics: Atlas*

Lists regions and countries defined in the atlas tag map.

Attributes

list="{regions, countries}"
Select what type of objects to list.

```
  <b>Available regions</b><br />
<emit source='atlas' list='regions'>
&_.name;<br />
</emit>
  <b>Available regions</b><br />

Africa<br />
```

```
Arab States<br />

Asia<br />

Europe<br />

North America<br />

Oceania<br />

South America<br />

World<br />
```

&_.name;
Provided by *Graphics: Atlas*
The name of the region/country

<emit source="cimg"></emit>

Provided by module: *Graphics: Image converter*

Entitybased version of <*cimg*>. Takes the same attributes as `<cimg>`.

&_.data;
Provided by *Graphics: Image converter*
Returns the imagedata given through other sources, like databases through entities.

&_.file-size;
Provided by *Graphics: Image converter*
Returns the image's file size.

&_.src;
Provided by *Graphics: Image converter*
Returns the path to the indata file.

&_.type;
Provided by *Graphics: Image converter*
Returns the image's content-type.

&_.xsize;
Provided by *Graphics: Image converter*
Returns the width of the image.

&_.ysize;
Provided by *Graphics: Image converter*
Returns the height of the image.

<emit source="dir"></emit>

Provided by module: *Tags: Dir emit source*

This plugin is used to generate directory listings. The directory module must be added to use these entities. This plugin is only available in the directory template.

Attributes

directory="path"
Apply the listing to this directory.

thumbnail-size="number"
Sets the size of the thumbnail. Defaultsize is 60 pixels. The size is set in proportion to the image's longest side, e.g. if the height of the image is longer than it's width, then the thumbnail will be 60 pixels high. The shortest side will be shown in proportion to the longest side.

thumbnail-format="imageformat"
Set the output format for the thumbnail. Default is *png*. All imageformats that <*cimg*> handles can be used to produce thumbnails.

strftime="strftime string" (%Y-%m-%d)
Format the date according to this string. Default is the isotime format (%Y-%m-%d), which will return (Year(four characters)-month(two characters)-day(two characters)), e.g. 2000-11-22. See the attribute *strftime* in <*date*> for a full listing of available formats.

glob="glob-pattern1[,glob-pattern2,...]"
Only show files matching the glob-pattern.

type="glob-pattern1[,glob-pattern2,...]"
Only show files which content-type matches the glob-pattern.

sort-order="{alpha, dwim, modified, size, type}" (dwim)
Sort the files and directories by this method.

alpha	Sort files and directories alphabetically.
dwim	Sort files and directories by "Do What I (want) Method". In many methods numeriacal sorts fail as the number '10' often appears before '2'. This method sorts numerical characters first then alphabetically, e.g. 1foo.html, 2foo.html, 10foo.html, foo1.html, foo2.html, foo10.html.
modified	Sort files by modification date.
size	Sort files by size.
type	Sort files by content-type.

sort-reversed
Reverse the sort order.

&_.atime;
Provided by *Tags: Dir emit source*
Returns the date when the file was last accessed.

&_.atime-iso;
Provided by *Tags: Dir emit source*
Returns the date when the file was last accessed. Uses isotime (%Y-%m-%d).

&_.atime-unix;
Provided by *Tags: Dir emit source*
Returns the date when the file was last accessed. Uses unixtime.

&_.dirname;
Provided by *Tags: Dir emit source*
Returns the directoryname.

&_.filename;

Provided by *Tags: Dir emit source*
Returns the filename.

&_.filesize;
Provided by *Tags: Dir emit source*
Returns a file's size in bytes. Directories get the size "-2".

&_.mode;
Provided by *Tags: Dir emit source*
Returns file permission rights represented binary, e.g. "r-xr-xr-x".

&_.mode-int;
Provided by *Tags: Dir emit source*
Returns file permission rights represented by integers. When encoded to binary this represents what is shown when using the Unix command "ls -l" or as shown using _.mode, e.g. "16749".

&_.mtime;
Provided by *Tags: Dir emit source*
Returns the date when the file was last modified.

&_.mtime-iso;
Provided by *Tags: Dir emit source*
Returns the date when the file was last modified. Uses isotime (%Y-%m-%d).

&_.mtime-unix;
Provided by *Tags: Dir emit source*
Returns the date when the file was last modified. Uses unixtime.

&_.name;
Provided by *Tags: Dir emit source*
Returns the name of the file or directory.

&_.path;
Provided by *Tags: Dir emit source*
Returns the path to the file or directory.

&_.real-dirname;
Provided by *Tags: Dir emit source*
Returns the directory of the real file in the filesystem.

&_.real-filename;
Provided by *Tags: Dir emit source*
Returns the path to the real file in the filesystem.

&_.size;
Provided by *Tags: Dir emit source*
Returns a file's size in kb(kilobytes).

&_.thumbnail;
Provided by *Tags: Dir emit source*
Returns the image associated with the file's content-type or directory.

&_.type;
Provided by *Tags: Dir emit source*
Returns the file's content-type.

&_.type-img;
Provided by *Tags: Dir emit source*
Returns the internal Roxen name of the icon representating the directory or the file's content-type, e.g. internal-gopher-menu for a directory-folder or internal-gopher-text for a HTML-file.

&_.vfs;
Provided by *Tags: Dir emit source*
Returns the name of the virtual filesystem that keeps the file.

&_.vfs-root;
Provided by *Tags: Dir emit source*
Returns the root directory of the virtual filesystem that keeps the file.

&_.x-size;
Provided by *Tags: Dir emit source*
Returns the width of the image.

&_.y-size;
Provided by *Tags: Dir emit source*
Returns the height of the image.

<emit source="fonts"></emit>

Provided by module: *Tags: RXML 2 tags*

Prints available fonts. This plugin makes it easy to list all available fonts in Roxen WebServer.

Attributes

type="{ttf, all}"
Which font types to list. ttf means all true type fonts, whereas all means all available fonts.

&_.copyright;
Provided by *Tags: RXML 2 tags*
Font copyright notice. Only available for true type fonts.

&_.expose;
Provided by *Tags: RXML 2 tags*
The preferred list name. Only available for true type fonts.

&_.family;
Provided by *Tags: RXML 2 tags*
The font family name. Only available for true type fonts.

&_.format;
Provided by *Tags: RXML 2 tags*
The format of the font file, e.g. ttf.

&_.full;
Provided by *Tags: RXML 2 tags*
The full name of the font. Only available for true type fonts.

&_.name;
Provided by *Tags: RXML 2 tags*
Returns a font identification name.
This example will print all available ttf fonts in gtext-style.

```
<emit source='fonts' type='ttf'>
  <gtext font='&_.name;'>&_.expose;</gtext><br />
</emit>
```

&_.path;
Provided by *Tags: RXML 2 tags*
The location of the font file.

&_.postscript;
Provided by *Tags: RXML 2 tags*
The fonts postscript identification. Only available for true type fonts.

&_.style;
Provided by *Tags: RXML 2 tags*
Font style type. Only available for true type fonts.

&_.trademark;
Provided by *Tags: RXML 2 tags*
Font trademark notice. Only available for true type fonts.

&_.version;
Provided by *Tags: RXML 2 tags*
The version of the font. Only available for true type fonts.

<emit source="js-dynamic-popup"></emit>

Provided by module: *Tags: Javascript Support*

Creates a dynamic load popup. This plugin creates a link to a dynamic loaded popup. The content of the popup will be loaded from the specified url.

Before this tag can be used a layer with the same name as this popup must be created with the `<js-dynamic-popup-div>` tag.

In order to use this tag the components; `CrossPlatform.js`, `Popup.js` and `DynamicLoading.js` must be included in the page with the `<js-include>` tag.

Note that dynamic loaded layers don't work with Netscape 6 browsers. Please see the comment in the beginning of the `DynamicLoading.js` component file for more information.

Attributes

src="url"
The page that should be loaded inte the popup.

name="string"
The name of the popup.

props="javascript object name" (default_props)
The name of the javascript PopupProperties object that is created by the tag. This object contains various properties for the popup. PopupProperties is defined in the *Popup.js* component and takes two arguments: x, and y offsets from the target event for positioning of the popup at a desired location.

There are some methods available in the object to set properties:

setHideDelay
The time in ms it takes before the popup is hidden when the mouse leaves the popup (default is 300 ms).

setHide2ndClick
If the popups parent is clicked a second time, the popup will be hidden if this method was called.

setParentRightOffset
The x offset from the parent popups right border. This offset will only be used if the popup has a parent popup i.e. not at the top level. This offset overrides the x_offset.

setParentBottomOffset
The y offset from the parent popups bottom border. This offset will only be used if the popup has a parent popup i.e. not at the top level. This offset overrides the y_offset.

setPageX
Sets the popup to this absolute x coordinate.

setPageY
Sets the popup to this absolute y coordinate.

An example that loads index.xml into a popup layer when the link is clicked.

```
<js-include file='CrossPlatform.js'/>
<js-include file='Popup.js'/>
<js-include file='DynamicLoading.js'/>

<js-dynamic-popup-div name='popup'/>

<emit source='js-dynamic-
popup' name='popup' src='index.xml'>
  <a href='javascript:void(0);' onClick='&_.event;'
>Show</a>
</emit>
```

&_.event;
Provided by *Tags: Javascript Support*
The javascript event.

<emit source="js-hide-popup"></emit>

Provided by module: *Tags: Javascript Support*

Creates a link event to hide popups. This plugin can be used in hierarchical menues on those links that are not popups, i.e. direct links on the same level as other links that leads to a popup.

```
<js-include file='CrossPlatform.js'/>
<js-include file='Popup.js'/>

<style><js-insert name='style'/></style>
<js-insert name='div'/>

<emit source='js-hide-popup'>
  <a href='index.xml' ::='&_.args;'>Hide</a>
</emit><br />
<js-popup label='Show'>
    ...
</js-popup>
```

&_.args;
Provided by *Tags: Javascript Support*
The javascript event arguments.

<emit source="languages"></emit>

Provided by module: *Preferred Language Analyzer*

Outputs language descriptions. It will output information associated to languages, such as the name of the language in different languages. A list of languages that should be output can be provided with the langs attribute. If no such attribute is used a generated list of present languages will be used. If such a list could not be generated the list provided in the Preferred Language Analyzer module will be used.

Attributes

langs

Should be a comma seperated list of language codes. The languages associated with these codes will be emitted in that order.

&_.code;
Provided by *Preferred Language Analyzer*
The language code.

&_.confurl;
Provided by *Preferred Language Analyzer*
A URL which makes the language the used one by altering the roxen cookie.

&_.en;
Provided by *Preferred Language Analyzer*
The language name in english.

&_.local;
Provided by *Preferred Language Analyzer*
The language name as written in the language itself.

&_.localized;
Provided by *Preferred Language Analyzer*
The language name as written in the currently selected language.

&_.preurl;
Provided by *Preferred Language Analyzer*
A URL which makes this language the used one by altering prestates.

<emit source="ldap"></emit>

Provided by module: *Tags: LDAP tags*

Use this source to search LDAP directory for information. The result will be available in variables named as the LDAP entries attribute.

Attributes

server="LDAP URL" (Server URL)
Connection LDAP URL. If omitted the *Default server URL* will be used.

search-filter="search filter"
Filter of an LDAP search operation. If used that this value rewrites the corresponding part of URL.

basedn="base DN"
Base DN of an LDAP search operation. If used that this value rewrites the corresponding part of URL.

search-scope="search scope"
Scope of an LDAP search operation. If used that this value rewrites the corresponding part of URL.

password="user password"
User password for connection to the directory server. If omitted the default will be used.

<emit source="path"></emit>

Provided by module: *Tags: RXML 2 tags*

Prints paths. This plugin traverses over all directories in the path from the root up to the current one.

Attributes

path="string"
Use this path instead of the document path

trim="string"
Removes all of the remaining path after and including the specified string.

skip="number"
Skips the 'number' of slashes ('/') specified, with beginning from the root.

skip-end="number"
Skips the 'number' of slashes ('/') specified, with beginning from the end.

&_.name;
Provided by *Tags: RXML 2 tags*
Returns the name of the most recently traversed directory.

&_.path;
Provided by *Tags: RXML 2 tags*
Returns the path to the most recently traversed directory.

<emit source="sources"></emit>

Provided by module: *Tags: RXML 2 tags*

Provides a list of all available emit sources.

&_.source;
Provided by *Tags: RXML 2 tags*
The name of the source.

<emit source="sql"></emit>

Provided by module: *Tags: SQL tags*

Use this source to connect to and query SQL databases for information. The result will be available in variables named as the SQL columns.

Attributes

host="database"
Which database to connect to, usually a symbolic name set in the *SQL Databases* module. If omitted the default database will be used.

query="SQL statement"
The actual SQL-statement.

<emit source="values"></emit>

Provided by module: *Tags: RXML 2 tags*

Splits the string provided in the values attribute and outputs the parts in a loop. The value in the values attribute may also be an array or mapping.

Attributes

values="string, mapping or array"

An array, mapping or a string to be splitted into an array.

split="string" (NULL)
The string the values string is splitted with. Supplying an empty string results in the string being split between every single character.

advanced="{lines, words, chars}"
If the input is a string it can be splitted into separate lines, words or characters by using this attribute.

case="{upper, lower}"
Changes the case of the value.

trimwhites
Trims away all leading and trailing white space charac-hters from the values.

from-scope="name"
Create a mapping out of a scope and give it as indata to the emit.

&_.index;
Provided by *Tags: RXML 2 tags*
The index of this mapping entry, if input was a mapping

&_.value;
Provided by *Tags: RXML 2 tags*
The value of one part of the splitted string

Database Tags

The database tags interact with SQL databases or LDAP directories. They can input/output information used in creating everything from simple tables or forms, interactive graphical reports to complete web applications.

The LDAP directory tag <*ldap*> interacts with LDAP directories as well as LDAP accessible directories like Novell NDS or Microsoft Active Directory.

The database tags are almost always combined with other RXML tags. Together with for instance <*vform*>, the <*sqlquery*> tag can be used to put data collected from the form into a database.

<*sqltable*> together with <*diagram*> provides real-time diagrams and with <*tablify*> nice looking tables can be dynamically generated.

The <*emit*> plugins <*emit sql*> or <*emit ldap*> can also be used to search and list information from a SQl database or LDAP directory.

Each database tag needs to know which database it should connect to. This is specified by the *host*-attribute which usually is a symbolic name for the database that the administrator has configured in the *SQL Databases* module. It is also possible to specify the database host, user and password directly in the tags, but this is not recommended.

<ldap/>

Provided by module: *Tags: LDAP tags*

Executes a LDAP operation, but doesn't do anything with the result.The `<ldap>` tag is mostly used for LDAP operation that change the contents of the directory, for example *add* or *modify*.

Attributes

server="LDAP URL" (Server URL)
Connection LDAP URL. If omitted the *Default server URL* will be used.

password="password"
User password for connection to the directory server. If omitted the default will be used.

dn="distinguished name"
Distinguished name of object.

op="add,delete,modify,replace"
The actual LDAP operation.
Note that *op='modify'* will change only the attributes given by the *attr* attribute.

attr="attribute_name1:[(attribute_value1[,...])][,attribute_name2,...]"
The actual values of attributes, for example:

```
(sn:'Zappa'),(mail:'hello@nowhere.org','athell@pandemonium.com')
```

parser
If specified, the content of *attr* will be parsed by the RXML parser. This is useful if the operation is to be built dynamically.

<sqlquery/>

Provided by module: *Tags: SQL tags*

Executes an SQL query, but doesn't do anything with the result. This is mostly used for SQL queries that change the contents of the database, for example INSERT or UPDATE.

Attributes

host="database"
Which database to connect to, usually a symbolic name set in the *SQL Databases* module. If omitted the default database will be used.

query="SQL statement"
The actual SQL-statement.

parse
If specified, the query will be parsed by the RXML parser. Useful if you wish to dynamically build the query. This attribute is deprecated and will have no effect if the servers compatibility level is above 2.1.

mysql-insert-id="variable"
Set the given variable to the insert id used by Mysql for auto-incrementing columns. Note: This is only available with Mysql.

<sqltable/>

Provided by module: *Tags: SQL tags*

Creates an HTML or ASCII table from the results of an SQL query.

Attributes

ascii
Create an ASCII table rather than an HTML table. Useful for interacting with <*diagram*> and <*tablify*>.

host="database"
Which database to connect to, usually a symbolic name set in the *SQL Databases* module. If omitted the default database will be used.

query="SQL statement"
The actual SQL-statement.

parse
If specified, the query will be parsed by the RXML parser. Useful if you wish to dynamically build the query.

Programming Tags

Programming tags can be used for advanced RXML such as making web applications. There are also tags of interest to module programmers. For those interested in combining programming with RXML, the *<?pike ?>* and *<?perl ?>* processing instruction tags enables embedded Pike and Perl code in RXML pages (see also the Script chapter in this manual).

<cache></cache>

Provided by module: *Tags: RXML 2 tags*

This tag caches the evaluated result of its contents. When the tag is encountered again in a later request, it can thus look up and return that result without evaluating the content again.

Nested `<cache>` tags are normally cached separately, and they are also recognized so that the surrounding tag doesn't cache their contents too. It's thus possible to change the cache parameters or completely disable caching of a certain part of the content inside a `<cache>` tag.

Note! This implies that any RXML tags that surrounds the inner `<cache>` tag(s) won't be cached.

Besides the value produced by the content, all assignments to RXML variables in any scope are cached. I.e. an RXML code block which produces a value in a variable may be cached, and the same value will be assigned again to that variable when the cached entry is used.

When the content is evaluated, the produced result is associated with a key that is built by taking the values of certain variables and other pieces of data, which thus are what the cache depends on. The arguments "variable", "key" and "profile" lets you specify the cache dependencies.

Note! It is easy to create huge amounts of cached values if the cache parameters are chosen badly. E.g. to depend on the contents of the form scope is typically only acceptable when combined with a fairly short cache time, since it's otherwise easy to fill up the memory on the server simply by making many requests with random variables.

The cache can be shared between all `<cache>` tags with identical content, which is typically useful in `<cache>` tags used in templates included into many pages. The drawback is that cache entries stick around when the `<cache>` tags change and that the cache won't be persistent (see below). Only shared caches have any effect if the RXML pages aren't compiled and cached.

If the page is compiled to p-code which is saved, and the cache is not shared, and there is no timeout on it, then the produced cache entries are also saved, and the cache is thus persistent. In this case it's thus especially important to limit the number of alternative cache entries.

Compatibility note: If the compatibility level of the site is lower than 2.2 and there are no "variable" or "profile" arguments, the cache depends on the contents of the form scope and the path of the current page (i.e. page.path). This is often a bad policy since it's easy for a client to generate many cache entries.

Attributes

variable="string"

This is a comma-separated list of variables and scopes that the cache should depend on. The value can be an empty string, which is useful to only disable the default dependencies in compatibility mode.

Since it's important to keep down the size of the cache, this should typically be kept to only a few variables with a limited set of possible values, or else the cache should have a timeout.

key="string"

Use the value of this attribute directly in the key. This attribute mainly exist for compatibility; it's better to use the "variable" attribute instead.

It is an error to use "key" together with "propagate", since it wouldn't do what you'd expect: The value for "key" would not be reevaluated when an entry is chosen from the cache, since the nested, propagating `<cache>` isn't reached at all then.

profile="string"

A comma-separated list to choose one or more profiles from a set of preconfigured cache profiles. Which cache profiles are available depends on the RXML parser module in use; the standard RXML parser currently has none.

shared

Share the cache between different instances of the `<cache>` with identical content, wherever they may appear on this page or some other in the same server. See the tag description for details about shared caches.

nocache

Do not cache the content in any way. Typically useful to disable caching of a section inside another cache tag.

propagate

Propagate the cache settings to the surrounding `<cache>` tag, if there is any. Useful to locally add depends to a cache without introducing a new cache level. If there is no surrounding `<cache>` tag, this argument is ignored.

nohash

If the cache is shared, then the content won't be made part of the cache key. Thus the cache entries can be mixed up with other `<cache>` tags.

not-post-method

By adding this attribute all HTTP requests using the POST method will be unaffected by the caching. The result

will be calculated every time, and the result will not be stored in the cache. The contents of the cache will however remain unaffected by the POST request.

flush-on-no-cache
If this attribute is used the cache will be flushed every time a client sends a pragma no-cache header to the server. These are e.g. sent when shift+reload is pressed in Netscape Navigator.

years="number"
Add this number of years to the time this entry is valid.

months="number"
Add this number of months to the time this entry is valid.

weeks="number"
Add this number of weeks to the time this entry is valid.

days="number"
Add this number of days to the time this entry is valid.

hours="number"
Add this number of hours to the time this entry is valid.

beats="number"
Add this number of beats to the time this entry is valid.

minutes="number"
Add this number of minutes to the time this entry is valid.

seconds="number"
Add this number of seconds to the time this entry is valid.

<?cdata ?>

Provided by module: *Tags: RXML 2 tags*

The content is inserted as a literal. I.e. any XML markup characters are encoded with character references. The first whitespace character (i.e. the one directly after the "cdata" name) is discarded.

This processing instruction is just like the <![CDATA[ ]]> directive but parsed by the RXML parser, which can be useful to satisfy browsers that does not handle <![CDATA[ ]]> correctly.

<cgi/>

Provided by module: *Scripting: CGI scripting support*

Runs a CGI script and inserts the result into the page.

Attributes

script="file"
Path to the CGI file to be executed.

cache="number" (0)
The number of seconds the result may be cached. If the attribute is given, but without any value or "0" as value it will use 60 seconds.

<clear-session/>

Provided by module: *Session tag module*

Clear a session from all its content.

Attributes

id="string"
The key that identifies the session.

<crypt></crypt>

Provided by module: *Tags: RXML 2 tags*

Encrypts the contents as a Unix style password. Useful when combined with services that use such passwords.

Unix style passwords are one-way encrypted, to prevent the actual clear-text password from being stored anywhere. When a login attempt is made, the password supplied is also encrypted and then compared to the stored encrypted password.

Attributes

compare="string"
Compares the encrypted string with the contents of the tag. The tag will behave very much like an *<if>* tag.

```
  <crypt compare="LAF2kkMr6BjXw">Roxen</crypt>
<then>Yepp!</then>
<else>Nope!</else>

Yepp!
```

<debug/>

Provided by module: *Tags: RXML 2 tags*

Helps debugging RXML-pages as well as modules. When debugging mode is turned on, all error messages will be displayed in the HTML code.

Attributes

on
Turns debug mode on.

off
Turns debug mode off.

toggle
Toggles debug mode.

showid="string"
Shows a part of the id object. E.g. showid="id->request_headers".

werror="string"
When you have access to the server debug log and want your RXML page to write some kind of diagnostics message or similar, the werror attribute is helpful.

This can be used on the error page, for instance, if you'd want such errors to end up in the debug log:

```
<debug werror='File &page.url; not found!
(linked from &client.referrer;)'/>
```

<dice></dice>

Provided by module: *Tags: Additional RXML tags*

Simulates a D&D style dice algorithm.

Attributes

type="string" (D6)
Describes the dices. A six sided dice is called 'D6' or '1D6', while two eight sided dices is called '2D8' or 'D8+D8'. Constants may also be used, so that a random number between 10 and 20 could be written as 'D9+10' (excluding 10 and 20, including 10 and 20 would be 'D11+9'). The character 'T' may be used instead of 'D'.

<eval></eval>

Provided by module: *Tags: RXML 2 tags*

Postparses its content. Useful when an entity contains RXML-code. `<eval>` is then placed around the entity to get its content parsed.

<force-session-id/>

Provided by module: *Session tag module*

Forces a session id to be set. The heuristics is as follows: If the RoxenUserID cookie is set, do nothing. Otherwise redirect to the same page but with a prestate containing a session key set. If now both the RoxenUserID cookie and the session prestate is set, redirect back to the same page without any session prestate set. RoxenUserID is automatically set by the HTTP protocol module. Look at the servers ports tab to enable this feature.

```
<force-session-id/>
<if variable='client.session'>
  RXML code that uses <ent>client.session</
ent>, e.g. <tag>session</tag><tag>/session</tag>.
</if>
```

<fsize/>

Provided by module: *Tags: RXML 2 tags*

Prints the size of the specified file.

Attributes

file="string"
Show size for this file.

<gauge></gauge>

Provided by module: *Tags: RXML 2 tags*

Measures how much CPU time it takes to run its contents through the RXML parser. Returns the number of seconds it took to parse the contents.

Attributes

define="string"
The result will be put into a variable. E.g. define="var.gauge" will put the result in a variable that can be reached with var.gauge.

silent
Don't print anything.

timeonly
Only print the time.

resultonly
Only print the result of the parsing. Useful if you want to put the time in a database or such.

<maketag></maketag>

Provided by module: *Tags: RXML 2 tags*

This tag creates tags. The contents of the container will be put into the contents of the produced container.

Attributes

name="string"
The name of the tag that should be produced. This attribute is required for tags, containers and processing instructions, i.e. for the types 'tag', 'container' and 'pi'.

```
  <maketag name='one' type='tag'></maketag>
<maketag name='one' type='tag' noxml='noxml'></
maketag>
  &lt;one /&gt;
&lt;one&gt;
```

noxml
Tags should not be terminated with a trailing slash. Only makes a difference for the type 'tag'.

type="{tag, container, pi, comment, cdata}"
What kind of tag should be produced. The argument 'Pi' will produce a processing instruction tag.

```
  <maketag type='pi' name='PICS'>l gen true r (n 0
s 0 v 0 l 2)</maketag>
  &lt;?PICS l gen true r (n 0 s 0 v 0 l 2)?&gt;
```

```
  <maketag type='comment'>Menu starts here</
maketag>
  &lt;!--Menu starts here--&gt;
```

```
<maketag type='comment'>Debug: &form.res; &var.sql;
</maketag>
```

```
  <maketag type='cdata'>Exact   words</maketag>
  &lt;![CDATA[Exact   words]]&gt;
```

<attrib></attrib>

Provided by module: *Tags: RXML 2 tags*

Inside the maketag container the container `<attrib>` is defined. It is used to add attributes to the produced tag. The contents of the attribute container will be the attribute value. E.g.

Attributes

```
  <eval>
<maketag name="replace" type="container">
 <attrib name="from">A</attrib>
 <attrib name="to">U</attrib>
 MAD
</maketag>
</eval>
```

```
 MUD
```

name="string"
The name of the attribute.

<nocache></nocache>

Provided by module: *Tags: RXML 2 tags*

Avoid caching of a part inside a `<cache>` tag. This is the same as using the `<cache>` tag with the nocache argument.

Note that when a part inside a `<cache>` tag isn't cached, it implies that any RXML tags that surround the `<nocache>` tag inside the `<cache>` tag also aren't cached.

<nooutput></nooutput>

Provided by module: *Tags: RXML 2 tags*

The contents will not be sent through to the page. Side effects, for example sending queries to databases, will take effect.

<noparse></noparse>

Provided by module: *Tags: RXML 2 tags*

The contents of this container tag won't be RXML parsed.

<?noparse ?>

Provided by module: *Tags: RXML 2 tags*

The content is inserted as-is, without any parsing or quoting. The first whitespace character (i.e. the one directly after the "noparse" name) is discarded.

<page-size/>

Provided by module: *Tags: Page sizer*

`<page-size>` calculates the size of a page, including inline images, and gives estimates of the time it will take to download the page. All information is shown in a box.

Attributes

page="path"
Calculate size and downloadtime for another page than the current.

include="summary,details,dltime,suggestions"
What information to present.

speeds="28.8,56.0,64.0,256.0,384.0,1024.0"
Show the time it will take to download a page using the specified speed(s) in kbit/s.

```
<page-size
 page='../'
 include='summary,details,dltime,suggestions'
 speeds='28.8,56.0,64.0,256.0,384.0,1024.0'
/>
```

<?perl ?>

Provided by module: *Scripting: Perl support*

This processing intruction tag allows for evaluating Perl code directly in the document.

Note: Read the installation and configuration documentation in the Administration manual to set up the Perl support properly. If the correct parameters are not set the Perl code might not work properly or security issues might arise.

There is also a `<perl>...</perl>` container tag available.

<?pike ?>

Provided by module: *Scripting: Pike tag*

This processing intruction tag allows for evaluating Pike code directly in the document.

Note: With this tag, users are able to run programs with the same right as the server. This is a serious security hasard.

When the pike tag returns, the contents of the output buffer is inserted into the page. It is not reparsed with the RXML parser.

Use entities within the Pike code, scope.variable is handled like scope.variable in RXML.

Note: It is still possible to use the `<pike>...</pike>` container tag, though it behaves exactly as it did in Roxen 2.0 and earlier and the functionality mentioned below does not apply to it. The use of the container tag is deprecated.

Below is a list of special helper functions and constructs which are only available within the `<?pike ?>` tag.

Attributes

write

write(string fmt, mixed ... args) is a helper function. It formats a string in the same way as printf and appends it to the output buffer. If given only one string argument, it's written directly to the output buffer without being interpreted as a format specifier.

flush
flush() is a helper function. It returns the contents of the output buffer and resets it.

rxml
rxml(string rxmlcode) is a helper function. It parses the string with the RXML parser.

"//O ..." or "/*O ... */"
Pike comment with an 'O' (the letter, not the number) as the very first character treats the rest of the text in the comment as output text that's written directly to the output buffer.

"//X ..." or "/*X ... */"
A Pike comment with an 'X' as the very first character treats the rest of the text in the comment as RXML code that's executed by the RXML parser and then written to the output buffer.

#include "..."
An #include preprocessor directive includes the specified file.

#inherit "..."
An #inherit preprocessor directive puts a corresponding inherit declaration in the class that's generated to contain the Pike code in the tag, i.e. it inherits a specified file from the Roxen filesystem.

```
<?pike
  //O <pre>
  int first = 1;
  for( var.counter=100; var.counter>1; var.counter-
-,first=0 )
  {
    if( !first )
    {
      //X &var.counter; bottles of beer on the wall
      //O
    }
    //X &var.counter; bottles of beer on the wall
    //X &var.counter; bottles of beer
    //O take one down, pass it around
  }
  //O one bottle of beer on the wall
  //O one bottle of beer
  //O take it down, pass it around
  //O no bottles of beer on the wall
  //O </pre>
?>
```

<session></session>

Provided by module: *Session tag module*

Creates a session bound scope. The session is identified by a session key, given as argument to the session tag. The session key could be e.g. a key generated by &unique-id;, transported by form variables.

Attributes

id="string"
The key that identifies the session. Could e.g. be a name, an IP adress or a cookie.

life="number" (900)
Determines how many seconds the session is guaranteed to persist on the server side. Values over 900 means that the session variables will be stored in a disk based database when they have not been used within 900 seconds.

force-db
If used, the session variables will be immediatly written to the database. Normally, e.g. when not defined, session variables are only moved to the database when they have not been used for a while (given that they still have "time to live", as determined by the life attribute). This will increase the integrity of the session, since the variables will survive a server reboot, but it will also decrease performance somewhat.

scope="name" (session)
The name of the scope that is created inside the session tag.

<set-max-cache/>

Provided by module: *Tags: RXML 2 tags*

Sets the maximum time this document can be cached in any ram caches.

Default is to get this time from the other tags in the document (as an example, *<if supports>* sets the time to 0 seconds since the result of the test depends on the client used.

You must do this at the end of the document, since many of the normal tags will override this value.

Attributes

years="number"
Add this number of years to the time this page was last loaded.

months="number"
Add this number of months to the time this page was last loaded.

weeks="number"
Add this number of weeks to the time this page was last loaded.

days="number"
Add this number of days to the time this page was last loaded.

hours="number"
Add this number of hours to the time this page was last loaded.

beats="number"
Add this number of beats to the time this page was last loaded.

minutes="number"
Add this number of minutes to the time this page was last loaded.

seconds="number"
Add this number of seconds to the time this page was last loaded.

<trace></trace>

Provided by module: *Tags: RXML 2 tags*

Executes the contained RXML code and makes a trace report about how the contents are parsed by the RXML parser.

<writefile></writefile>

Provided by module: *Tags: Writefile*

Writes uploaded or direct content to a file. You can either use an upload form or write the container content directly into a file. The ownership of any newly created file is determined by the directory it is placed into.

Additional functionality includes removal or renaming of already existing files. This container tag will set the truth value depending on success or failure of the requested operation.

Attributes

filename="string"
Specifies the virtual filename to be created or operated on (relative to the current directory, or to the root of the virtual filesystem).

chroot="string"
Specifies the virtual root directory (sandbox) all file operations are contained under.

from="string"
Specifies the type=file form field variable which uploaded the file to be written. If this attribute is omitted, the container content is what will be written instead. Given the example below, the parameter *from=wrapupafile* should be specified.

```
<form method='post'
   enctype='multipart/form-data'>
 <input type='file' name='wrapupafile' />
 <input type='submit' value='Upload file' />
</form>
File uploaded:
   <insert scope='form'
     variable='wrapupafile.filename'/>
```

append
Append to the file instead of replacing it.

mkdirhier
Create the directory hierarchy needed to store the file if needed.

remove
Causes the specified filename or directory to be removed.

moveto="string"
Causes the specified filename to be moved to this new location.

max-size="integer"
Specifies the maximum upload file size in bytes which is accepted.

max-height="integer"
The maximum imageheight in pixels which is accepted.

max-width="integer"
The maximum imagewidth in pixels which is accepted.

min-height="integer"
The minimum imageheight in pixels which is accepted.

min-width="integer"
The minimum imagewidth in pixels which is accepted.

accept-type="string"
Comma separated list of file types which are accepted, currently supported types are jpeg, png and gif; the check is performed on the file content, not on the file extension.

Script

When web server tasks that are too complex to do in RXML, or take too much time to run as RXML, need to be carried out, there are primarily two ways of doing this: writing scripts, or writing extension modules for the server. (Writing extension modules for the server is an advanced option, and not covered here. See the Roxen Programmer Manual for more information about writing Roxen modules.)

In the first few years of web server programming, scripts were launched through by means of a mechanism known as CGI (Common Gateway Interface), and this is still the most general and basic form of using script. Since then, some scripting languages have been given special support -- for the purpose of greater flexibility and improved performance -- in several web servers. In Roxen, there are primarily three scripting languages that have this kind of special support: Pike, Perl and Java.

Another scripting language is Javascript, but it is different from most other scripting languages in use on the Web in the sense that it is a client-side, not a server-side, language. Java (which should not be confused with Javascript) is both a server-side and a client-side language.

CGI

Any script language for which there is a script interpreter installed in your system (at a location where Roxen can access it) can be run as a CGI (Common Gateway Interface) script. The traditional form for this is to write them as Unix scripts, with a first line of

```
#! /path/to/the/interpreter
```

followed by the actual script. An example in Perl (assuming the Perl interpreter is installed as /usr/bin/perl, which is the common place for it on many Unix systems) might look like this:

```
#! /usr/bin/perl print "Content-type: text/
html\r\n"; print
"\r\n"; print "Current time according to the server
: ",
scalar(localtime), "\n";
```

On Unix, we can use the traditional login shell as a script interpreter, if we want to:

```
#! /bin/sh
echo 'Content-type: text/html\r'
echo '\r'
echo "Current time according to the server: `date`"
```

See the CGI section in the Roxen Programmer Manual for more information about CGI scripts.

Pike

Pike is the 'native' language of Roxen, and is directly supported by the web server. This also makes it the most powerful language in Roxen, since it allows direct interfacing with the server's native API.

It is possible that the Roxen administrator will prefer not to allow ordinary users to take advantage of this functionality, since it can compromise web server security; but even if this is the case for your site, Pike can still be available through CGI, if the Pike interpreter has been installed in a publicly available place in your system.

See the section about Pike in the Roxen Programmer Manual for more details about Pike scripts.

Note! A special feature that might be enabled in your Roxen server is to allow Pike code in XML-style *Preprocessor Instructions*. This takes the form of <?pike _the_code_goes_here_ ?>, and is described in the section about Programming Tags in this manual.

Perl

Perl is one of the most common scripting languages in the web world, and can be run either in the traditional way through CGI, or by the special Perl support present in Roxen. Either of these features need to be enabled in your server to allow you to use Perl scripts.

See the section about Perl in the Roxen Programmer Manual for more details.

Note! A special feature that might be enabled in your Roxen server is to allow Perl code in XML-style *Preprocessor Instructions*. This takes the form of <?perl _the_code_goes_here_ ?>, and is described in the section about Programming Tags in this manual.

Java

Java exists both as a server-side language and a client-side language. For client-side use of Java, the server just passes the Java program on to the browser, without doing anything particular with the code. For server-side use, there are primarily two forms: Java Servlets, and Java Server Pages. A Java Servlet is a special script that acts a bit like a miniature web server for part of the processing of a request to the server, while Java Server Pages (JSP) works more like traditional CGI scripts.

In Roxen, Java Server Pages is handled through a special Java Servlet, which must be enabled in the server to be available to page authors.

See also the section about Java in the Roxen Programmer Manual for more details.

Javascript

Javascript is a client-side language (whose real name, by the way, is nowadays *ECMA script*), and should not be confused with Java. You can write Javascript inside your RXML pages, and Roxen will handle it like other web servers: i.e. ignore it, and passing it on as ordinary text to the browser. In other words, it has essentially nothing with the web server to do, and is exclusively a matter between the page author and the page visitor's browser.

Keep in mind, however, that it will pass through an XML parsing stage just as any other text in an RXML page, so symbols that have special meaning in XML, such as `<` and `&` may have to be quoted to reach the browser unscathed.

Consult a book on Javascript/ECMA-script for more information about how to use that language, e.g "JavaScript, The Definitive Guide, 4 th ed." by David Flanagan (2001).

Javascript Support

The *Tags: Javascript Support* module adds some RXML features, mostly to create popup windows with RXML contents inside. These tags are quite tightly dependent on one another, and hence most useful to demonstrate with examples. To fully use their potential, you need to be intimately familiar with javascript (or ECMAscript, to be true to the proper name of the language) programming - we can reccommend David Flanagan's excellent *JavaScript: The Definitive Guide, 3rd Edition*, for instance.

We could, for example, create a popup window with a few convenience links like this:

```
<js-include file="CrossPlatform.js"/>
<js-include file="Popup.js"/>
<style>
  <js-insert name="style"/>
</style>
<js-insert name="div" jswrite="jswrite"/>
<js-insert name="javascript1.2"/>
<js-popup args-variable="my-popup" oy="15" ox="0">
<gtext font="franklin_gothic_demi" scale="0.40"
fgcolor="black" black="black">Teleport to</gtext>
<obox>
<a href="http://www.roxen.com/">roxen</a><br />
<a href="http://docs.roxen.com/">docs</a><br />
<a href="http://pike.roxen.com/">pike</a>
</obox>
</js-popup>

<a ::="&form.my-popup;">Popup and away!</a>
```

<js-dynamic-popup-div/>

Provided by module: *Tags: Javascript Support*

Creates a dynamic popup div/layer tag. This tag creates a div or layer tag depending on the browser used. Use this tag together with the *<emit js-dynamic-popup>* tag.

Attributes

name="string"
The name of the div/layer.

z-index="number" (1)
The z index for the div/layer.

<js-external></js-external>

Provided by module: *Tags: Javascript Support*

The tag creates an external javascript file containing the content of the tag in javascript *document.write* fashion. When the browser loads the document the external file will write the contents back to the document.

This tag can be used to exclude some parts of the document, replacing each with a reference to an external file. By sharing fragments that are identical and occur in several pages, page loading time can be shortened. The gain depends on the cache policy of the browser.

An important issue is that if the server uses a https port then the browsers won't cache the external files and using this tag will not result in increased performance.

Note that it is not possible to put a `<js-insert>` tag inside this tag because that tag uses a filter module to insert its content. Other RXML tags should work, however.

Note that this tag is not compatible with serverside persistent caching because the generated external file is only stored in the servers memory. A restart of the server will clear the information about the external files but the referenses to them do still exists in pages that are cached persistent.

<js-include/>

Provided by module: *Tags: Javascript Support*

Includes a javascript component. Required before using some of the other tags in the javascript support.

Attributes

file="component name"
The component to include. May be one of `CrossPlatform.js`, `DragDrop.js`, `DynamicLoading.js`, `Popup.js` or `Scroll.js`.

<js-insert/>

Provided by module: *Tags: Javascript Support*

Inserts a javascript support string.

Attributes

name="string"
The name of the javascript support string to insert.

jswrite
The output will be turned into a javascript tag and written to the page with the *document.write* funtion. This is useful for compatibility with browsers that has disabled javascript.

<js-popup></js-popup>

Provided by module: *Tags: Javascript Support*

This tag creates a popup of its content and returns a link that activates the popup if the cursor hovers over the link.

This tag generates some javascript support strings that have to be inserted into the page with the `<js-insert>` tag. The strings are: *div* and *style*.

The components `CrossPlatform.js` and `Popup.js` must be included with the `<js-include>` tag in order to use this tag.

All arguments exept those given below are also transferred to the link.

Attributes

href="url" (javascript:void(0);)

If you didn't give a href argument of your own to the link, the javascript "do nothing" statement is filled in for you.

label="string"

The link text. If omitted no link will be returned, useful in combination with the args-variable argument.

props="javascript object name" (default_props)

The name of the javascript PopupProperties object that is created by the tag. This object contains various properties for the popup. PopupProperties is defined in the *Popup.js* component and takes two arguments: x, and y offsets from the target event for positioning of the popup at a desired location.

There are some methods available in the object to set properties:

setHideDelay

The time in ms it takes before the popup is hidden when the mouse leaves the popup (default is 300 ms).

setHide2ndClick

If the popups parent is clicked a second time, the popup will be hidden if this method was called.

setParentRightOffset

The x offset from the parent popups right border. This offset will only be used if the popup has a parent popup i.e. not at the top level. This offset overrides the x_offset.

setParentBottomOffset

The y offset from the parent popups bottom border. This offset will only be used if the popup has a parent popup i.e. not at the top level. This offset overrides the y_offset.

setPageX

Sets the popup to this absolute x coordinate.

setPageY

Sets the popup to this absolute y coordinate.

A small example that uses a props object:

```
<js-include file='CrossPlatform.js'/>
<js-include file='Popup.js'/>

<style><js-insert name='style'/></style>
<js-insert name='div'/>

<script language='javascript'>
  popup_props = new PopupProperties(15, 0);
  popup_props.setHide2ndClick();
  popup_props.setHideDelay(500);
</script>

<js-popup props='popup_props' label='popup'>
  <h1>This is a popup!</h1>
</js-popup>
```

args-variable="RXML variable name"

Arguments to the generated anchor tag will be stored in this variable. This argument is useful if the target to the popup should be an image, see the example below.

```
<js-include file='CrossPlatform.js'/>
<js-include file='Popup.js'/>

<style><js-insert name='style'/></style>
<js-insert name='div'/>

<js-popup args-variable='popup-args'>
  <h1>This is a popup!</h1>
</js-popup>

<gtext ::='&form.popup-args;'>popup</gtext>
```

event="javascript event" (onMouseOver)

The javascript event that should trigger the popup.

z-index="number" (1)

The z index for the popup's div tag.

<js-write></js-write>

Provided by module: *Tags: Javascript Support*

Converts its content into javascript document.write. The output will be turned into a javascript tag and written to the page with the *document.write* funtion. This is useful for compatibility with browsers that has disabled javascript.

SSI Tags

SSI, Server Side Includes, are similar to RXML tags and have the advantage of being a standard supported by many web servers. It is thus possible to write pages using SSI that are portable to other web servers.

The downside of SSI is that it is in no way as flexible or powerful as RXML. The tags are placed within HTML comments, which makes it impossible to combine different SSI tags. However, it is possible to combine SSI tags with regular RXML tags.

Roxen WebServer does not presently implement all the SSI functionality that Apache supports.

<!–#config –>

Provided by module: *Tags: SSI support*

The config command is used to configure how things should be printed.

Attributes

errmsg="string"
Where msg is a message that is sent back to the client if an error occurs while parsing the SSI-tag.

sizefmt="{bytes, abbrev}"
The value sets the format to be used when displaying the size of a file. Bytes gives a count in bytes while abbrev gives a count in KB or MB, as appropriate.

timefmt="value"
The value is a string to be used when displaying SSI date output.

<!–#echo –>

Provided by module: *Tags: SSI support*

Prints a variable from the server or request.

Some of the most useful ones are "http referrer" (the page which contained the link to the current page), "last modified" (date of the file for this document), "remote user" (name of the user), and "remote addr" (IP number of the user/client machine).

Note that these variables are SSI-related. You cannot access them as RXML variables, nor use this tag to print RXML variables.

Attributes

var="sizefmt"
Print format for file sizes.

var="document name"
Name of the current document (= page). RXML counterpart: page.self.

```
<!--#echo var="document name" -->
index.html
```

var="document uri"
URI (URL) to the current page. RXML counterpart: page.url.

```
<!--#echo var="document uri" -->
/index.html
```

var="date local"
Time and date, in current time zone. RXML counterpart: `<date strftime="%c"/>`

var="date gmt"
Time and date, GMT time zone. RXML counterpart: `<date timezone="GMT" strftime="%c"/>`

var="last modified"
Last time this document was modified. RXML counterpart: `<modified/>`.

var="server software"
The web server software. RXML counterpart: roxen.version.

var="server name"
The web server name. RXML counterpart: roxen.domain.

var="remote host"
Name of client machine. RXML counterpart: client.host.

var="remote addr"
Numeric IP address of client machine. RXML counterpart: client.ip.

var="auth type"
Authentication type (typically Basic).

var="remote user"
Client user name.

var="http referrer"
URL of the referring page. RXML counterpart: client.referrer.

var="gateway interface"
Answers "CGI/1.1".

var="http cookie"
A list of the set cookies.

```
<!--#echo var="http cookie" -->
```

var="cookie"
A list of the set cookies. Same as http cookie.

var="http accept"
A list of the http accept formats.

```
<!--#echo var="http accept" -->
Unknown variable (http accept).
```

var="http user agent"
The user agent string. RXML counterpart: client.Fullname.

var="path translated"
Translated path.

var="query string unescaped"
The query string.

var="request method"
Request method (typically GET).

var="server protocol"
Protocol used for request.

var="server port"
Server's port number.

<!--#exec -->

Provided by module: *Tags: SSI support*

Executes a CGI script or shell command. This command has security implications and therefore, might not be available on all web sites.

Attributes

cgi="URL"
Path to the CGI script URL encoded. That is, a character can be quoted by % followed by its hex value. The CGI script is given the PATH_INFO and QUERY_STRING of the original request from the client. The variables available in `<!--#echo>` will be available to the script in addition to the standard CGI environment. If the script returns a Location header, then this will be translated into an HTML anchor.

cmd="path"
The server will execute the command using /bin/sh. The variables available in `<!--#echo>` will be available to the script.

<!--#flastmod -->

Provided by module: *Tags: SSI support*

This tag prints the last modification date of the specified file, subject to timefmt format specification used in the *<!--#config>* SSI tag.

Attributes

file="path"
Path to the file.

virtual="URL"
Path to the file URL encoded. That is, a character can be quoted by % followed by its hex value.

<!--#fsize -->

Provided by module: *Tags: SSI support*

Prints the size of the specified file, subject to the sizefmt format specification used in the `<!--#config>` SSI tag.

Attributes

file="path"
Path to the file.

virtual="URL"
Path to the file URL encoded. That is, a character can be quoted by % followed by its hex value.

<!--#include -->

Provided by module: *Tags: SSI support*

Insert a text from another file into the page.

Attributes

file="path"
The file as a path relative to the directory containing the current page. It cannot contain ../, nor can it be an absolute path.

virtual="URL"
The path to the file, URL encoded. That is, a character can be quoted by % followed by its hex value. The path may contain ../ and may be absolute, i e starting with a /.

<!--#printenv -->

Provided by module: *Tags: SSI support*

This tag outputs a listing of all existing variables and their values. Attributes won't be printed.

```
<pre><!--#printenv --></pre>
```

<!--#set -->

Provided by module: *Tags: SSI support*

Sets the value of a variable.

Attributes

var="value"
The name of the variable to set. Value sets the variables value.

Compatibility

This chapter is only applicable when upgrading from an older version of a Roxen product. The use of tags listed here is not recommended.

Upgrading from 1.3 or older

With the redesign of RXML many tags have been rewritten to better fit into the new RXML design, some have been obsolete because of redundency matters, to avoid security and similar problems or simply because better tags have been invented.

The RXML syntax in Roxen Challenger 1.3 and Roxen WebServer 2.2 isn't compatible. This means that old websites built with Roxen Challenger won't operate properly under Roxen WebServer 2.1. To ensure full functionality in Roxen WebServer 2.1 one of these items has to be fulfilled.

- Make the RXML code compatible with XML by adding /'s (slash) to all the tags, enclose all tag attributes with "'s (quotationmark) and remove tags like `<formoutput>` and `<sqloutput>`. Read more about XML *here*.
- Add the appropriate 'Compatibility modules' to the site. The old RXML code on the site will now function exactly as it did in Roxen Challenger 1.3. Ask an administrator to enable the appropriate modules.

<counter/>

Provided by module: *Graphics: Counter*

<cset></cset>

Provided by module: *Tags: RXML 2 tags*

Sets a variable with its content. This is deprecated in favor of using the <set></set> construction.

Attributes

variable="name"
The variable to be set.

quote="{html, none}"
How the content should be quoted before assigned to the variable. Default is html.

<if eval></if>

Provided by module: *Tags: RXML 2 tags*

Deprecated due to non-XML compliancy. The XML standard says that attribute-values are not allowed to contain any markup. The `<if eval>` tag was deprecated in Roxen 2.0.

```
<!-- If eval statement -->
<if eval="<foo>">x</if>

<!-- Compatible statement -->
<define variable="var.foo" preparse="preparse"><foo
/></define>
<if sizeof="var.foo">x</if>
```

A similar but more XML compliant construct is a combination of `<define variable>` and an apropriate `<if>` plugin.

Security

Htaccess is a system for handling access control to your pages. It works by placing an *.htaccess* file in a directory, which contains the access control lists for that directory. It is possible to get fine grained security and to configure exactly who can view which pages. The *.htaccess support* module must be enabled for htaccess to work.

It is possible to distinguish users either by the IP address or domain name of their computer or by letting them authenticate with a user name and password. The user name and password is by default compared via an *authentication* module. That usually means that users are authenticated by the operating systems authentication system.

If you want to have password protected pages usable by users that are not handled by the current Authentication module you can create your own database of users by creating *.htpasswd* and *.htgroup* files.

htaccess is a standard supported by many web servers. The access control you have built with htaccess should therefore be portable to other web servers.

.htaccess

A `.htaccess` file consists of lines containing directives. Apart from the Limit; directive, all directives have the form

```
directive argument(s)
```

where *argument(s)* is one or more arguments. The directives supported are:

AuthUserFile
Use this user and password file to authenticate users. Typically, the AuthUserFile is called *.htpasswd*

AuthGroupFile
Use this group file, which contains a database of which groups users are member of. Typically, the AuthGroupFile is called *.htgroup*, if used.

AuthName
Set the authentication realm, which can be any name you choose. The name will be used to tell browsers how to *label* user authentications within a session, so that the browsers can automatically repeat passwords the user has already entered when accessing new pages with the same access requirements.

Redirect
Redirect all accesses for pages in the directory to this URL.

ErrorFile
Show this page in case the requested page could not be found, maybe because the user did not have permission to view it.

Then there is the `<Limit>` container tag. The attributes are the HTTP method(s) that access should be limited to, *GET*, *PUT*, *POST* or *HEAD*. The contents of the tag are access control directives, one directive on each line. Possible directives are:

allow|deny from URL
Allow or deny access to users from a DNS domain or IP number. *www.roxen.com* means the computer while *.roxen.com* means all computers on the domain *roxen.com*. The same way *194.52.202.3* means the computer while *194.52.* means the net starting with *194.52*

require user|group user(s)|group(s)
Allow access only for the named user(s) or group(s).

require valid-user
Allow access to any user present in the AuthUserFile or Authentication module.

satisfy all|any
Decide what happens if both *require* and *allow* rules are present; *all* indicates that the user must satisfy both kinds of requirements, while *any* means that it is enough that the user satisfies either kind.

order deny,allow|allow,deny|mutual-failure
The *order* rules decides how to prioritize deny and allow rules. If the order is set to *deny,allow*, deny rules will be processed before allow rules. With *allow,deny*, allows will be processed before denies, and with *mutual-failure*, hosts allowed by any *allow* rule will be allowed, and other hosts denied. *Deny,allow* is the default.

The rule evaluation does not stop until all rules have been processed, so the earlier a rule is processed, the lower priority is has in determining access. This only matters when different rules contradict each other, for instance when a wide-ranging deny rule forbids access to a certain domain, and an allow grants access to a smaller part of the domain.

Example

A typical *.htaccess* file would look something like this:

```
AuthUserFile /home/frotz/.htpasswd
AuthGroupFile /home/frotz/.htgroup
AuthName MyTestDomain
AuthType Basic

<Limit PUT HOST HEAD>
require user frotz
</Limit>

<Limit GET>
allow from all
</Limit>
```

The *.htaccess* file above would allow everyone to GET documents in the directory, but all other kinds of access would be restricted to the user frotz, and expect this user to login with the password listed for frotz in the *.htpasswd* file in the home directory of the user frotz.

.htgroup

The format of the group file is straightforward, one line per group, with the line containing the group name, followed by a colon, followed by the users in the group. Users are separated by commas.

In other words, an *.htgroup* file can look like this:

```
all:frotz,gnusto
admins:frotz
```

with one line per group.

.htpasswd

The format of the password file is straightforward, one line per user, with the line containing the user name, followed by a colon, followed by the user's password encrypted with the standard Unix password encryption. *<crypt>* can be used to encrypt such a password.

In other words, an *.htpasswd* can look like this:

```
frotz:taeWr6tbTZKO6
gnusto:jKXVnZH6eXR7
```

with one line for each user.